

**ΥΠΟΕΡΓΟ: ΥΠΟΕΡΓΟ 2 «ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΤΑΡΤΙΣΗΣ, ΑΝΑΠΤΥΞΗΣ ΔΕΞΙΟΤΗΤΩΝ, ΕΝΔΥΝΑΜΩΣΗΣ, ΠΙΣΤΟΠΟΙΗΣΗΣ - ΥΛΟΠΟΙΗΣΗ ΜΕ ΙΔΙΑ ΜΕΣΑ, ΕΠΙΜΟΡΦΩΣΗ ΑΠΟ ΤΟ ΕΚΔΔΑ» του Έργου «SUB4. Αναβάθμιση των δεξιοτήτων του ανθρώπινου δυναμικού του Δημόσιου Τομέα» με κωδικό ΟΠΣ ΤΑ 5150174**

**της Δράσης 16972 ΤΑΑ**

**TΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:**

**ΕΞ ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ ΣΤΗ   
ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΩΝ LINUX (ΜΕΡΟΣ Β)**

**ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ**

**Κωδικός εκπαιδευτικού υλικού:**

**Κωδικός Πιστοποίησης προγράμματος: XXX**

**ΥΠΟΕΡΓΟ: : ΥΠΟΕΡΓΟ 2 «ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΤΑΡΤΙΣΗΣ, ΑΝΑΠΤΥΞΗΣ ΔΕΞΙΟΤΗΤΩΝ, ΕΝΔΥΝΑΜΩΣΗΣ, ΠΙΣΤΟΠΟΙΗΣΗΣ - ΥΛΟΠΟΙΗΣΗ ΜΕ ΙΔΙΑ ΜΕΣΑ, ΕΠΙΜΟΡΦΩΣΗ ΑΠΟ ΤΟ ΕΚΔΔΑ» του Έργου «SUB4. Αναβάθμιση των δεξιοτήτων του ανθρώπινου δυναμικού του Δημόσιου Τομέα» με κωδικό ΟΠΣ ΤΑ 5150174 της Δράσης 16972 ΤΑΑ**

**TΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:**

**ΕΞ ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ ΣΤΗ**

**ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΩΝ LINUX (ΜΕΡΟΣ Β)**

**ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ**

**Μέλη Ομάδας**

**Συντονίστρια:**

**Χαραλαμπία Δουλή**

**Συγγραφείς:**

**Χρήστος Γρομπανόπουλος**

**Δημήτριος Καραπιπέρης**

**Αλέξανδρος Γιοχάλας**

**Αξιολογητές/τριες:**

**Μαρία Τράκα**

**Αντώνιος Στεφάνου**

στο EXCEL

Η ΧΡΗΣΗ ΤΟΥ

Power Query

ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ

Περιεχόμενα

[ΚΕΦΑΛΑΙΟ 1: Επίβλεψη και διαχείριση διεργασιών 7](#_Toc204859905)

[**1.1.** Βασικές έννοιες των διεργασιών 7](#_Toc204859906)

[1.1.1. Βασικά Χαρακτηριστικά μιας διεργασίας. 7](#_Toc204859907)

[1.1.2. Ομάδες διεργασιών και σύνοδοι 8](#_Toc204859908)

[**1.2.** Παρουσίαση των διεργασιών με την εντολή ps 9](#_Toc204859909)

[1.2.1. Το εικονικό σύστημα αρχείων /proc 12](#_Toc204859910)

[**1.3.** Εκτέλεση και διαχείριση διεργασιών στο παρασκήνιο 13](#_Toc204859911)

[1.3.1. Έναρξη διεργασίας στο παρασκήνιο 13](#_Toc204859912)

[1.3.2. Διαχείριση διεργασιών στο παρασκήνιο με την εντολή jobs 14](#_Toc204859913)

[1.3.3. Μεταφορά διεργασιών μεταξύ προσκηνίου και παρασκηνίου 14](#_Toc204859914)

[1.3.4. Αναφορά σε εργασίες 15](#_Toc204859915)

[**1.4.** Σήματα διεργασιών 15](#_Toc204859916)

[1.4.1. Αποστολή σημάτων με την εντολή kill 15](#_Toc204859917)

[1.4.2. Βασικά Σήματα στο Linux 16](#_Toc204859918)

[1.4.3. Αποστολή σημάτων με την εντολή killall 16](#_Toc204859919)

[**1.5.** Προτεραιότητες διεργασιών 17](#_Toc204859920)

[1.5.1. Αλλαγή Προτεραιότητας σε Εκτελούμενη Διεργασία 17](#_Toc204859921)

[**1.6.** Παρακολούθηση και διαχείριση διεργασιών με την χρήση εργαλείων 17](#_Toc204859922)

[1.6.1. Παρακολούθηση και διαχείριση διεργασιών με την εντολή top 17](#_Toc204859923)

[1.6.2. Παρακολούθηση και διαχείριση διεργασιών με το System Monitor 19](#_Toc204859924)

[**1.7.** Άσκηση – Διαχείριση διεργασιών από την γραμμή εντολών 21](#_Toc204859925)

[1.7.1. Εργασία 1. Εμφάνιση πληροφοριών για τις διεργασίες 21](#_Toc204859926)

[1.7.2. Εργασία 2. Αποστολή σημάτων σε διεργασίες 23](#_Toc204859927)

[1.7.3. Εργασία 3. Παρακολούθηση της λειτουργίας του συστήματος με την χρήση της top 25](#_Toc204859928)

[1.7.4. Εργασία 4: Αλλαγή της προτεραιότητας εκτέλεσης μιας διεργασίας με nice/renice 26](#_Toc204859929)

[**1.8.** Ερωτήσεις αυτοαξιολόγησης 27](#_Toc204859930)

[1.8.1. Απαντήσεις στις ερωτήσεις αυτοαξιολόγησης 29](#_Toc204859931)

[ΚΕΦΑΛΑΙΟ 2: Διαχείριση Υπηρεσιών και Δαιμόνων 30](#_Toc204859932)

[**2.1.** Υπηρεσίες (Services) 30](#_Toc204859933)

[2.1.1. To init σύστημα 30](#_Toc204859934)

[2.1.2. Είδη init συστημάτων 31](#_Toc204859935)

[**2.2.** To systemd init σύστημα 32](#_Toc204859936)

[Τα κύρια πλεονεκτήματα του systemd είναι: 32](#_Toc204859937)

[2.2.1. Service units 33](#_Toc204859938)

[2.2.2. Target units 33](#_Toc204859939)

[2.2.3. Socket units 35](#_Toc204859940)

[2.2.4. Mount units 36](#_Toc204859941)

[**2.3.** Δαίμονες (Daemons) 38](#_Toc204859942)

[2.3.1. Η εντολή systemctl 38](#_Toc204859943)

[2.3.2. Προβλήματα κατά την εκκίνηση δαιμόνων 40](#_Toc204859944)

[2.3.3. Δικτυακές υπηρεσίες 42](#_Toc204859945)

[**2.4.** Διαφορές μεταξύ υπηρεσιών και δαιμόνων 43](#_Toc204859946)

[**2.5.** Άσκηση: Παρακολούθηση και έλεγχος δικτυακών υπηρεσιών και υπηρεσιών συστήματος 44](#_Toc204859947)

[2.5.1. Ενδεικτική επίλυση 44](#_Toc204859948)

[**2.6.** Ερωτήσεις αυτοαξιολόγησης 46](#_Toc204859949)

[2.6.1. Απαντήσεις 48](#_Toc204859950)

[ΚΕΦΑΛΑΙΟ 3: Διαχείριση και ασφάλιση του SSH – Ανάλυση και αποθήκευση των συμβάντων καταγραφής 49](#_Toc204859951)

[**3.1.** Σύνδεση σε απομακρυσμένο υπολογιστή με SSH 49](#_Toc204859952)

[3.1.1. Χρήση SSH για μεταφορά αρχείων 50](#_Toc204859953)

[3.1.2. Συμπεράσματα 50](#_Toc204859954)

[**3.2.** Ρύθμιση χρήσης κλειδιών (δημόσιου-ιδιωτικού) για την πιστοποίηση της ταυτότητας του χρήστη χωρίς την εισαγωγή κωδικού. 51](#_Toc204859955)

[3.2.1. Δημιουργία ζεύγους κλειδιών 51](#_Toc204859956)

[3.2.2. Μεταφορά του δημοσίου κλειδιού στον απομακρυσμένο υπολογιστή 51](#_Toc204859957)

[3.2.3. Ασφάλεια και πλεονεκτήματα 52](#_Toc204859958)

[**3.3.** Περιορισμός της απευθείας σύνδεσης ως διαχειριστής και απενεργοποίηση του ελέγχου πιστοποίησης ταυτότητας με χρήση κωδικού 53](#_Toc204859959)

[3.3.1. Περιορισμός σύνδεσης του διαχειριστή του συστήματος 53](#_Toc204859960)

[3.3.2. Απενεργοποίηση πιστοποίησης μέσω κωδικού πρόσβασης 53](#_Toc204859961)

[3.3.3. Οφέλη αυτών των μέτρων 54](#_Toc204859962)

[**3.4.** Χρήση της υπηρεσίας fail2ban για την προστασία των συνδέσεων 54](#_Toc204859963)

[3.4.1. Βασικές ρυθμίσεις 55](#_Toc204859964)

[3.4.2. Έλεγχος της κατάστασης 56](#_Toc204859965)

[3.4.3. Πλεονεκτήματα 57](#_Toc204859966)

[**3.5.** Άσκηση – Ρύθμιση και ασφάλιση του SSH 57](#_Toc204859967)

[3.5.1. Ενδεικτική επίλυση 58](#_Toc204859968)

[**3.6.** Aρχιτεκτονική καταγραφής συμβάντων με το journald 58](#_Toc204859969)

[**3.7.** Εύρεση και ερμηνεία καταγραφών για την επίλυση προβλημάτων 60](#_Toc204859970)

[**3.8.** Διατήρηση των εγγραφών συμβάντων 63](#_Toc204859971)

[**3.9.** Ρύθμιση συγχρονισμού ώρας και ζώνης ώρας με το NTP 63](#_Toc204859972)

[**3.10.** Ασκηση - Επισκόπηση, διατήρηση του συστήματος καταγραφής (system journal) και ρύθμιση της ώρας συστήματος μέσω του πρωτοκόλλου NTP 66](#_Toc204859973)

[3.10.1. Ενδεικτική επίλυση 66](#_Toc204859974)

[**3.11.** Ερωτήσεις αυτοαξιολόγησης 67](#_Toc204859975)

[3.11.1. Απαντήσεις 68](#_Toc204859976)

[ΚΕΦΑΛΑΙΟ 4: Διαχείριση Δικτύων 70](#_Toc204859977)

[**4.1.** Βασικές έννοιες δικτύων 70](#_Toc204859978)

[4.1.1. Το δικτυακό μοντέλο TCP/IP 70](#_Toc204859979)

[4.1.2. Ονοματοδοσία των θυρών δικτύου 71](#_Toc204859980)

[4.1.3. Δίκτυα IPv4 72](#_Toc204859981)

[4.1.4. Δίκτυα IPv6 76](#_Toc204859982)

[4.1.5. Ονόματα υπολογιστών και διευθύνσεις IP 79](#_Toc204859983)

[4.1.6. Ερωτήσεις ενότητας 80](#_Toc204859984)

[**4.2.** Επιβεβαιώνοντας τις ρυθμίσεις δικτύου 81](#_Toc204859985)

[4.2.1. Συλλέγοντας πληροφορίες για τις διεπαφές δικτύου 81](#_Toc204859986)

[4.2.2. Έλεγχος συνδεσιμότητας μεταξύ υπολογιστών 83](#_Toc204859987)

[4.2.3. Έλεγχος και επιδιόρθωση προβλημάτων δρομολόγησης 85](#_Toc204859988)

[4.2.4. ΈΛΕΓΧΟΣ ΚΑΙ ΕΠΙΔΙΟΡΘΩΣΗ ΣΕ ΘΥΡΕΣ ΚΑΙ ΥΠΗΡΕΣΙΕΣ 87](#_Toc204859989)

[**4.3.** ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ 88](#_Toc204859990)

[4.3.1. Ολοκλήρωση 91](#_Toc204859991)

[**4.4.** ΡΥΘΜΙΣΗ ΤΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΗ ΓΡΑΜΜΗ ΕΝΤΟΛΩΝ 91](#_Toc204859992)

[4.4.1. ΕΝΝΟΙΕΣ ΤΟΥ NETWORKMANAGER 91](#_Toc204859993)

[4.4.2. ΕΜΦΑΝΙΖΟΝΤΑΣ ΠΛΗΡΟΦΟΡΙΕΣ ΔΙΚΤΥΟΥ 92](#_Toc204859994)

[4.4.3. ΠΡΟΣΘΕΤΟΝΤΑΣ ΜΙΑ ΝΕΑ ΣΥΝΔΕΣΗ 93](#_Toc204859995)

[4.4.4. ΕΛΕΓΧΟΝΤΑΣ ΤΙΣ ΣΥΝΔΕΣΕΙΣ ΔΙΚΤΥΟΥ 93](#_Toc204859996)

[4.4.5. ΑΛΛΑΖΟΝΤΑΣ ΤΙΣ ΡΥΘΜΙΣΕΙΣ ΣΥΝΔΕΣΕΩΝ ΔΙΚΤΥΟΥ 94](#_Toc204859997)

[4.4.6. ΑΦΑΙΡΩΝΤΑΣ ΜΙΑ ΣΥΝΔΕΣΗ ΔΙΚΤΥΟΥ 96](#_Toc204859998)

[4.4.7. ΔΙΚΑΙΩΜΑΤΑ ΑΛΛΑΓΩΝ ΣΤΙΣ ΡΥΘΜΙΣΕΙΣ ΔΙΚΤΥΟΥ 96](#_Toc204859999)

[4.4.8. ΣΥΝΟΨΗ ΤΩΝ ΕΝΤΟΛΩΝ 97](#_Toc204860000)

[**4.5.** ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ 97](#_Toc204860001)

[ΡΥΘΜΙΣΗ ΤΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΗ ΓΡΑΜΜΗ ΕΝΤΟΛΗΣ 97](#_Toc204860002)

[**4.6.** ΕΠΕΞΕΡΓΑΣΙΑ ΑΡΧΕΙΩΝ ΡΥΘΜΙΣΕΩΝ ΔΙΚΤΥΟΥ 102](#_Toc204860003)

[4.6.1. ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΑΡΧΕΙΩΝ ΡΥΘΜΙΣΕΩΝ ΣΥΝΔΕΣΕΩΝ 102](#_Toc204860004)

[4.6.2. ΑΛΛΑΖΟΝΤΑΣ ΤΙΣ ΡΥΘΜΙΣΕΙΣ ΔΙΚΤΥΟΥ 105](#_Toc204860005)

[**4.7.** ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ ΕΠΕΞΕΡΓΑΣΙΑ ΑΡΧΕΙΩΝ ΡΥΘΜΙΣΕΩΝ ΔΙΚΤΥΟΥ 105](#_Toc204860006)

[**4.8.** ΡΥΘΜΙΖΟΝΤΑΣ ΟΝΟΜΑΤΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΝ ΕΠΙΛΥΣΗ ΟΝΟΜΑΤΩΝ 110](#_Toc204860007)

[4.8.1. ΑΛΛΑΖΟΝΤΑΣ ΤΟ ΟΝΟΜΑ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ 110](#_Toc204860008)

[4.8.2. ΡΥΘΜΙΖΟΝΤΑΣ ΤΗΝ ΕΠΙΛΥΣΗ ΟΝΟΜΑΤΩΝ 110](#_Toc204860009)

[**4.9.** ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ ΡΥΘΜΙΣΗ ΟΝΟΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΠΙΛΥΣΗΣ ΟΝΟΜΑΤΩΝ 112](#_Toc204860010)

[ΚΕΦΑΛΑΙΟ 5: Εγκατάσταση και ενημέρωση πακέτων λογισμικού – Ρύθμιση της πρόσβασης στο σύστημα των αρχείων 117](#_Toc204860011)

[**5.1.** Εγκατάσταση και ενημέρωση πακέτων λογισμικού 117](#_Toc204860012)

[5.1.1. Βασικές έννοιες των πακέτων λογισμικού 117](#_Toc204860013)

[5.1.2. Εργαλεία διαχείρισης πακέτων λογισμικού 118](#_Toc204860014)

[5.1.3. Εγκατάσταση / απεγκατάσταση / ενημέρωση πακέτων λογισμικού 119](#_Toc204860015)

[5.1.4. Αναζήτηση πακέτων λογισμικού – Προβολή λιστών λογισμικού 121](#_Toc204860016)

[5.1.5. Ομάδες (group) πακέτων λογισμικών 122](#_Toc204860017)

[5.1.6. Ρυθμίσεις του DNF: Διαχείριση αποθετηρίων και παραμετροποίηση 123](#_Toc204860018)

[**5.2.** Άσκηση: Εγκατάσταση και ενημέρωση πακέτων λογισμικού 125](#_Toc204860019)

[5.2.1. Εργασία 1: Εγκατάσταση / απεγκατάσταση / ενημέρωση πακέτων λογισμικού 125](#_Toc204860020)

[5.2.2. Εργασία 2: Αναζήτηση και δημιουργία λιστών με πακέτα λογισμικού 127](#_Toc204860021)

[5.2.3. Εργασία 3: Πληροφορίες για πακέτο λογισμικού 128](#_Toc204860022)

[5.2.4. Εργασία 4: Διαχείριση ομάδων πακέτων λογισμικού 130](#_Toc204860023)

[5.2.5. Εργασία 5: Ρυθμίσεις του DNF και των αποθετηρίων 131](#_Toc204860024)

[**5.3.** Ρύθμιση της πρόσβασης στο σύστημα των αρχείων 132](#_Toc204860025)

[5.3.1. Συσκευές τύπου Block 132](#_Toc204860026)

[5.3.2. Συστήματα αρχείων (filesystems) 137](#_Toc204860027)

[5.3.3. Προσάρτηση συσκευών αποθήκευσης 139](#_Toc204860028)

[5.3.4. Το σύστημα Λογικών Τόμων (LVM) 140](#_Toc204860029)

[5.3.5. Αναζήτηση αρχείων στον δενδρικό κατάλογο με χρήση της locale 141](#_Toc204860030)

[5.3.6. Αναζήτηση αρχείων στο δενδρικό κατάλογο με την χρήση της find 142](#_Toc204860031)

[**5.4.** Άσκηση: Ρύθμιση της πρόσβασης στο σύστημα αρχείων 147](#_Toc204860032)

[5.4.1. Εργασία 1: Λειτουργίες διαχείρισης Συσκευών αποθήκευσης 147](#_Toc204860033)

[5.4.2. Εργασία 2: Λειτουργίες του LVM 149](#_Toc204860034)

[5.4.3. Εργασία 3: Αναζήτηση αρχείων και καταλόγων με την χρήση των εντολών find και locate 151](#_Toc204860035)

[ΒΙΒΛΙΟΓΡΑΦΙΑ 153](#_Toc204860036)

# Επίβλεψη και διαχείριση διεργασιών

Οι βασικές σχεδιαστικές αρχές του λειτουργικού συστήματος Linux περιλαμβάνουν τη δυνατότητα να εξυπηρετεί πολλαπλούς χρήστες ταυτόχρονα (multi-user), καθώς και την ικανότητα κάθε χρήστη να εκτελεί πολλές διεργασίες συγχρόνως (multi-process). Αυτές οι δυνατότητες καθιστούν το Linux ένα ισχυρό και ευέλικτο σύστημα για καθημερινή χρήση, ειδικά σε περιβάλλοντα όπου απαιτείται η εκτέλεση πολλών εργασιών ταυτόχρονα.

Ο στόχος αυτού του κεφαλαίου είναι να εξηγήσει τις βασικές έννοιες που αφορούν τις διεργασίες στο Linux, να αναλύσει τη λειτουργία τους και να παρουσιάσει τις εντολές που χρησιμοποιούνται για την παρακολούθηση και διαχείριση αυτών.

## Βασικές έννοιες των διεργασιών

Τι είναι, λοιπόν, μία διεργασία; Στο Linux, μία διεργασία (process) είναι μια ανεξάρτητη οντότητα που δημιουργείται κάθε φορά που εκτελείται ένα πρόγραμμα ή μία εντολή. Αποτελεί τη βάση πάνω στην οποία λειτουργεί το λειτουργικό σύστημα, το οποίο αναλαμβάνει τη διαχείριση των διεργασιών, διασφαλίζοντας την ομαλή εκτέλεσή τους.

Ας πάρουμε για παράδειγμα έναν φυλλομετρητή (web browser). Ένα τέτοιο πρόγραμμα μπορεί να εκτελείται ταυτόχρονα πολλές φορές, είτε από τον ίδιο χρήστη είτε από διαφορετικούς. Για κάθε μία από αυτές τις εκτελέσεις, το λειτουργικό σύστημα δημιουργεί μία ή περισσότερες διακριτές διεργασίες, που λειτουργούν αυτόνομα, παρά το ότι εκτελούν το ίδιο πρόγραμμα.

Εκτός από τις διεργασίες που ξεκινούν οι χρήστες, το σύστημα δημιουργεί και άλλες διεργασίες που εξυπηρετούν λειτουργίες στο παρασκήνιο. Αυτές οι διεργασίες, γνωστές ως **system processes** (διεργασίες συστήματος), υποστηρίζουν υπηρεσίες όπως DNS servers, web servers ή άλλες κρίσιμες λειτουργίες του λειτουργικού συστήματος. Μάλιστα μπορεί οι διεργασίες αυτές μπορούν να ξεκινούν αυτόματα με την εκκίνηση του συστήματος. Αντίθετα, οι διεργασίες που δημιουργούνται από τους χρήστες ονομάζονται **user processes** (διεργασίες χρήστη).

### Βασικά Χαρακτηριστικά μιας διεργασίας.

Κάθε διεργασία που εκτελείται σε ένα λειτουργικό σύστημα Linux διαθέτει συγκεκριμένα χαρακτηριστικά και ιδιότητες που επιτρέπουν την αναγνώριση, τη διαχείριση και την παρακολούθησή της. Ας εξετάσουμε τα βασικά χαρακτηριστικά που την ορίζουν.

1. **Μοναδικό Αναγνωριστικό (PID): Κάθε διεργασία έχει έναν μοναδικό αριθμό αναγνώρισης, γνωστό ως PID (Process Identifier). Αυτός ο αριθμός εκχωρείται από το λειτουργικό σύστημα κατά τη δημιουργία της διεργασίας και παραμένει μοναδικός καθ' όλη τη διάρκεια ζωής της.**

**Για παράδειγμα, αν εκτελεστεί το ίδιο πρόγραμμα δύο φορές, κάθε εκτέλεση θα αντιπροσωπεύεται από ξεχωριστή διεργασία με διαφορετικό PID. Η μοναδικότητα του PID επιτρέπει στο σύστημα να παρακολουθεί και να διαχειρίζεται κάθε διεργασία ανεξάρτητα.**

1. **«Πατρική» Διεργασία και Ιεραρχία: Στο Linux, κάθε διεργασία δημιουργείται από μια άλλη, αποκαλούμενη πατρική διεργασία (parent process). Η μόνη εξαίρεση σε αυτόν τον κανόνα είναι η πρώτη διεργασία που έχει PID το μηδέν (0) και δημιουργείται κατά την εκκίνηση του συστήματος.**
   * **Στις παλαιότερες διανομές Linux, αυτή η πρώτη διεργασία ήταν η Init.**
   * **Στις νεότερες διανομές, έχει αντικατασταθεί από τη systemd, η οποία αποτελεί τον αρχικό πυρήνα διαχείρισης των διεργασιών του συστήματος.**

**Η κάθε πατρική διεργασία μπορεί να δημιουργήσει μία ή περισσότερες διεργασίες-παιδιά (child processes). Για την παρακολούθηση αυτής της σχέσης, κάθε διεργασία διατηρεί το PID της πατρικής της διεργασίας, γνωστό ως PPID (Parent PID). Αυτή η ιεραρχία διασφαλίζει την οργανωμένη και αποτελεσματική διαχείριση των διεργασιών.**

1. **Κατάσταση Διεργασίας: Κάθε διεργασία σε ένα λειτουργικό σύστημα Linux βρίσκεται ανά πάσα στιγμή σε μία από τις προκαθορισμένες καταστάσεις, οι οποίες περιγράφουν την τρέχουσα δραστηριότητά της. Οι βασικές καταστάσεις διεργασιών περιλαμβάνουν:**
   * **Εκτελείται (Running): Η διεργασία εκτελείται ενεργά από τον επεξεργαστή. Σε ένα σύστημα με πολλαπλές διεργασίες, κάθε στιγμή εκτελούνται μόνο αυτές που έχουν διατεθεί στον επεξεργαστή, ενώ οι υπόλοιπες βρίσκονται σε αναμονή.**
   * **Σε Αναμονή (Waiting): Η διεργασία περιμένει να ολοκληρωθεί κάποιο εξωτερικό γεγονός. Για παράδειγμα, μπορεί να περιμένει την ανάγνωση δεδομένων από τον σκληρό δίσκο ή την άφιξη δεδομένων από μια δικτυακή σύνδεση.**
   * **Σταματημένη (Stopped): Η διεργασία έχει σταματήσει προσωρινά από ενέργεια του χρήστη. Σε αυτή την κατάσταση, η διεργασία δεν εκτελείται, αλλά μπορεί να συνεχιστεί κατόπιν εντολής.**
   * **Ζόμπι (Zombie): Η διεργασία έχει ολοκληρώσει την εκτέλεσή της, αλλά το PID της παραμένει στη μνήμη μέχρι να το «συλλέξει» η πατρική της διεργασία. Οι ζόμπι διεργασίες δεν εκτελούν καμία εργασία, αλλά καταλαμβάνουν πόρους του συστήματος, και αν παραμείνουν πολλές, μπορεί να επηρεάσουν την απόδοση.**
   * **Αδρανής (Idle): Η διεργασία βρίσκεται σε αναστολή και δεν απαιτεί πόρους, παραμένοντας έτοιμη για ενεργοποίηση μόλις κριθεί απαραίτητο από το λειτουργικό σύστημα.**

### **Ομάδες διεργασιών και σύνοδοι**

Το λειτουργικό σύστημα Linux συχνά οργανώνει διεργασίες σε ομάδες όταν αυτές σχετίζονται μεταξύ τους ή συνεργάζονται για την εκτέλεση μέρους μιας πολύπλοκης λειτουργίας. Η ομαδοποίηση αυτή επιτρέπει στο λειτουργικό σύστημα να διαχειρίζεται πιο αποτελεσματικά τις διεργασίες, αντιμετωπίζοντας μια ομάδα ως ενιαία μονάδα.

**Ομάδες διεργασιών (Process Groups)**

Μια ομάδα διεργασιών (process group) αποτελείται από μία ή περισσότερες διεργασίες που συνεργάζονται για έναν κοινό σκοπό. Ένα χαρακτηριστικό παράδειγμα είναι οι διεργασίες που δημιουργούνται για τη λειτουργία μιας υπηρεσίας, όπως ενός web server. Ένα άλλο παράδειγμα είναι η εκτέλεση μιας σύνθετης εντολής στον φλοιό (shell) που χρησιμοποιεί διοχετεύσεις (pipes). Σε αυτό το πλαίσιο, μια ομάδα διεργασιών μπορεί να αναφέρεται και ως εργασία (job), καθώς όλες οι σχετικές διεργασίες εκτελούν τμήματα της ίδιας εργασίας.

Κάθε ομάδα διεργασιών έχει έναν μοναδικό αναγνωριστικό αριθμό, γνωστό ως PGID (Process Group ID). Συνήθως, το PGID μιας ομάδας αντιστοιχεί στο PID της αρχικής διεργασίας της ομάδας, γνωστής και ως leader διεργασία.

**Σύνοδοι (Sessions)**

Διάφορες ομάδες διεργασιών μπορούν να ενταχθούν σε μία σύνοδο (session), μια ανώτερη δομή που βοηθά το λειτουργικό σύστημα να οργανώνει και να απομονώνει διαφορετικές ομάδες διεργασιών. Η δημιουργία συνόδων παρέχει μεγαλύτερο έλεγχο και αποτρέπει παρεμβολές μεταξύ διεργασιών που ανήκουν σε διαφορετικές εργασίες ή υπηρεσίες.

Μια νέα σύνοδος δημιουργείται στις εξής περιπτώσεις:

**Διαδραστικός φλοιός (Interactive Shell):** Όταν ένας χρήστης συνδέεται στο σύστημα μέσω ενός διαδραστικού φλοιού, η αρχική διεργασία του φλοιού γίνεται ο ηγέτης της συνόδου. Η σύνοδος αυτή συνδέεται επίσης με ένα τερματικό ελέγχου.

**Υπηρεσία Δαίμονας (Daemon):** Όταν μια υπηρεσία τύπου δαίμονα ξεκινά, συχνά δημιουργεί μια δική της σύνοδο, ώστε να απομονωθεί από τις διεργασίες του χρήστη και του συστήματος. Περισσότερα για αυτού του τύπου τις υπηρεσίες θα παρουσιάσουμε σε επόμενο κεφάλαιο.

**Τερματικό (Terminal)**

Μια σύνοδος μπορεί να συνδέεται με ένα τερματικό ελέγχου, μια διεπαφή μέσω της οποίας ο χρήστης αλληλεπιδρά με το λειτουργικό σύστημα. Στα πρώτα συστήματα Unix, το τερματικό ελέγχου ήταν μια φυσική συσκευή, όπως μια κονσόλα. Στα σύγχρονα λειτουργικά συστήματα, το τερματικό είναι συνήθως μια εικονική συσκευή, γνωστή και ως ψευδο-τερματικό (pseudo-terminal). Αυτή η εικονική συσκευή παρέχεται από εφαρμογές τερματικού που προσομοιώνουν τη λειτουργία των παλιών φυσικών τερματικών.

## Παρουσίαση των διεργασιών με την εντολή ps

Η εντολή **ps** είναι το κύριο εργαλείο που χρησιμοποιείται για την επισκόπηση των διεργασιών που εκτελούνται στον υπολογιστή. Μέσω αυτής μπορούμε να δούμε ποια προγράμματα ή εντολές εκτελούνται, πώς σχετίζονται μεταξύ τους, και ποιοι πόροι χρησιμοποιούνται.

**Βασική Χρήση**

Η πιο απλή εκτέλεση της εντολής παρουσιάζει τις διεργασίες που τρέχουν στη σύνοδο του χρήστη στο τρέχον τερματικό:

[user@host ~]$ ps

PID TTY TIME CMD

13428 pts/0 00:00:00 bash

13596 pts/0 00:00:00 ps

* **PID**: Ο μοναδικός αριθμός ταυτότητας της διεργασίας.
* **TTY**: Το τερματικό από το οποίο ξεκίνησε η διεργασία.
* **TIME**: Ο χρόνος CPU που έχει χρησιμοποιηθεί από τη διεργασία.
* **CMD**: Το όνομα της εντολής που εκτελεί τη διεργασία.

**Πλήρης παρουσίαση με την επιλογή -f**

Για περισσότερες λεπτομέρειες σχετικά με τις διεργασίες, χρησιμοποιούμε την επιλογή -f:

[user@host ~]$ ps -f

UID PID PPID C STIME TTY TIME CMD

user 13428 13333 0 07:01 pts/0 00:00:00 bash

user 13604 13428 0 07:14 pts/0 00:00:00 ps -f

Η έξοδος περιλαμβάνει τα παρακάτω επιπλέον χαρακτηριστικά σε σχέση με την βασική χρήση:

* **UID:** Τον χρήστη στον οποίο ανήκει η διεργασία.
* **PPID:** Το PID της «πατρικής» διεργασίας που δημιούργησε την τρέχουσα.
* **C:** Το ποσοστό χρήσης του επεξεργαστή.
* **STIME:** Η ώρα εκκίνησης της διεργασίας.
* **TTY:** Το τερματικό εκτέλεσης.

Στο παράδειγμα παρατηρούμε ότι o φλοιός bash (PID 13428) είναι η «πατρική» διεργασία για την ps (PID 13604).

**Ιδιομορφία στο Συντακτικό της ps**

Η εντολή ps διαθέτει τρεις διαφορετικούς τρόπους παροχής επιλογών, οι οποίοι μπορούν να προκαλέσουν σύγχυση:

Α) Επιλογές UNIX: Προηγείται παύλα (-) και οι επιλογές μπορούν να ομαδοποιηθούν (π.χ. ps -aux).

Β) Επιλογές BSD: Δεν απαιτείται παύλα και οι επιλογές μπορούν επίσης να ομαδοποιηθούν (π.χ. ps aux).

Γ) Εκτεταμένες επιλογές GNU: Ξεκινούν με διπλή παύλα (--) και δεν ομαδοποιούνται (π.χ. ps --help).

Παρατηρήστε ότι οι συντακτικές αυτές παραλλαγές μπορούν να παράγουν διαφορετικά αποτελέσματα, ακόμη και αν οι εντολές μοιάζουν (π.χ. ps aux και ps -aux).

**Παρουσίαση όλων των διεργασιών με την επιλογή -e**

Αν θέλουμε να εμφανίσουμε όλες τις διεργασίες που εκτελούνται στο σύστημά μας, μπορούμε να χρησιμοποιήσουμε την επιλογή -e της εντολής ps. Η επιλογή αυτή εμφανίζει πλήρη λίστα διεργασιών ανεξάρτητα από το τερματικό ή τον χρήστη που τις εκτέλεσε. Για την εμφάνιση αναλυτικότερων πληροφοριών, μπορούμε να συνδυάσουμε την επιλογή -e με την επιλογή -f, η οποία προσθέτει περισσότερα πεδία στην έξοδο, όπως τον χρήστη, το αναγνωριστικό πατρικής διεργασίας και την ώρα εκκίνησης.

Επειδή τα αποτελέσματα της εντολής ps -e -f μπορεί να είναι εκτενή, μπορούμε να φιλτράρουμε την έξοδο για να εμφανιστούν μόνο οι πρώτες γραμμές, χρησιμοποιώντας την εντολή head -5:

UID PID PPID C STIME TTY TIME CMD

root 1 0 0 07:00 ? 00:00:01 /usr/lib/systemd/systemd rhgb --switched-root --system --deserialize 31

root 2 0 0 07:00 ? 00:00:00 [kthreadd]

root 3 2 0 07:00 ? 00:00:00 [rcu\_gp]

root 4 2 0 07:00 ? 00:00:00 [rcu\_par\_gp]

Από την παραπάνω έξοδο παρατηρούμε ότι:

* Ορισμένες διεργασίες δεν σχετίζονται με κανένα τερματικό (TTY = ?), γεγονός που υποδεικνύει ότι εκτελούνται στο παρασκήνιο.
* Οι διεργασίες με TTY = ? συνήθως ανήκουν στον χρήστη root και είναι κρίσιμες για τη λειτουργία του συστήματος.
* Η διεργασία με PID 1 είναι η systemd, η οποία είναι η πρώτη διεργασία που εκτελείται από το σύστημα κατά την εκκίνηση.
* Οι διεργασίες που δεν σχετίζονται με τερματικό και εκτελούνται στο παρασκήνιο, συχνά αποκαλούνται daemons. Θα εξεταστούν αναλυτικότερα σε επόμενο κεφάλαιο.

**Καθορισμός της μορφής εξόδου με την επιλογή -o**

Η εντολή ps μας επιτρέπει να προσαρμόσουμε τη μορφή της εξόδου, καθορίζοντας συγκεκριμένα πεδία που θέλουμε να εμφανιστούν. Αυτό γίνεται χρησιμοποιώντας την επιλογή -o, ακολουθούμενη από μία λίστα ονομάτων πεδίων.

Για να δούμε ποια πεδία είναι διαθέσιμα, χρησιμοποιούμε την εντολή:

ps L

Για παράδειγμα, αν θέλουμε να εμφανίσουμε μόνο το Process ID (PID), την εντολή που εκτελεί τη διεργασία (COMMAND) και την κατάσταση της διεργασίας (STATE), χρησιμοποιούμε:

[user@host ~]$ ps -o pid,comm,state

PID COMMAND S

13788 bash S

13926 vi T

13935 ps R

Από την παραπάνω έξοδο, παρατηρούμε:

* Η διεργασία ps βρίσκεται σε κατάσταση Running (R), δηλαδή εκτελείται ενεργά από τον επεξεργαστή.
* Η διεργασία bash βρίσκεται σε κατάσταση Sleeping (S), δηλαδή περιμένει κάποιο γεγονός για να συνεχίσει την εκτέλεση.
* Η διεργασία vi βρίσκεται σε κατάσταση Stopped (T), δηλαδή έχει σταματήσει προσωρινά λόγω σήματος.

Η στήλη STATE χρησιμοποιεί μονογράμματα για να παρουσιάσει την τρέχουσα κατάσταση κάθε διεργασίας. Στον παρακάτω πίνακα παρατίθενται οι πιο συνηθισμένες καταστάσεις διεργασιών:

|  |  |
| --- | --- |
| **Σύμβολο** | **Κατάσταση** |
| **R** | Εκτελείται (Running) |
| **S** | Αναμονή (Sleeping) |
| **T** | Σταματημένη (Stopped) |
| **Z** | Ζόμπι (Zombie) |
| **D** | Αναμονή σε μη διακοπτόμενη κατάσταση (Uninterruptible Sleep) |

**Ταξινομημένη εμφάνιση αποτελεσμάτων με την επιλογή --sort**

Μπορούμε να ταξινομήσουμε την εμφάνιση των διεργασιών που παρουσιάζει η εντολή ps με βάση κάποιο από τα πεδία τους, χρησιμοποιώντας την επιλογή --sort <όνομα\_πεδίου>. Επίσης, για να καθορίσουμε τη σειρά ταξινόμησης, μπορούμε να προσθέσουμε + ή - πριν από το όνομα του πεδίου, ώστε να ορίσουμε αύξουσα ή φθίνουσα σειρά αντίστοιχα.

Παρακάτω δίνεται ένα παράδειγμα χρήσης της επιλογής --sort για ταξινόμηση με βάση το πεδίο comm, όπου εμφανίζονται μόνο οι τελευταίες 10 γραμμές του αποτελέσματος:

[user@host ~]$ ps -ef --sort comm | tail -10

root 642 2 0 07:00 ? 00:00:00 [xfs-log/dm-0]

root 820 2 0 07:00 ? 00:00:00 [xfs-log/nvme0n1]

root 639 2 0 07:00 ? 00:00:00 [xfs-reclaim/dm-]

root 817 2 0 07:00 ? 00:00:00 [xfs-reclaim/nvm]

root 636 2 0 07:00 ? 00:00:00 [xfs\_mru\_cache]

root 644 2 0 07:00 ? 00:00:00 [xfsaild/dm-0]

root 822 2 0 07:00 ? 00:00:00 [xfsaild/nvme0n1]

root 635 2 0 07:00 ? 00:00:00 [xfsalloc]

root 859 2 0 07:00 ? 00:00:00 [xprtiod]

root 111 2 0 07:00 ? 00:00:00 [zswap-shrink]

**Επιπλέον Επιλογές της ps**

Η εντολή ps διαθέτει πολλές άλλες επιλογές που μπορούν να χρησιμοποιηθούν για την προσαρμογή της εξόδου. Παρακάτω παρουσιάζονται μερικές χαρακτηριστικές επιλογές:

|  |  |
| --- | --- |
| **Επιλογή** | **Λειτουργικότητα** |
| -p pidlist | Εμφανίζει πληροφορίες μόνο για συγκεκριμένο PID |
| -c command | Φιλτράρει διεργασίες που εκτελούν μια συγκεκριμένη εντολή |
| -t ttylist | Φιλτράρει διεργασίες με βάση τον τερματικό τους (TTY) |
| -u userlist | Εμφανίζει διεργασίες συγκεκριμένων χρηστών |

**Εντολή pidof**

Συχνά, χρειάζεται να εντοπίσουμε το PID μιας συγκεκριμένης εντολής που εκτελείται. Για αυτή την περίπτωση, η εντολή pidof μας δίνει την απάντηση άμεσα.

Παράδειγμα:

[user@host ~]$ pidof bash

2753

[user@host ~]$

### Το εικονικό σύστημα αρχείων /proc

Ο κατάλογος /proc είναι ένα ψευδο-σύστημα αρχείων που βρίσκεται σε όλα τα συστήματα Linux. Αν και δεν πρόκειται για πραγματικό σύστημα αρχείων, λειτουργεί ως σημείο πρόσβασης σε δεδομένα που παρέχει ο πυρήνας του συστήματος. Μέσω αυτού του καταλόγου, μπορούμε να ανακτήσουμε πολύτιμες πληροφορίες σχετικά με τις διεργασίες, το υλικό, τη μνήμη και διάφορα άλλα στοιχεία του συστήματος.

Για παράδειγμα, το αρχείο /proc/sys/kernel/hostname περιέχει το όνομα του υπολογιστή. Μπορούμε να το δούμε εκτελώντας την εξής εντολή:

[user@host ~]$ cat /proc/sys/kernel/hostname

host

Επιπλέον, ο κατάλογος /proc περιέχει έναν φάκελο για κάθε διεργασία που εκτελείται στο σύστημα. Κάθε φάκελος έχει το PID της αντίστοιχης διεργασίας ως όνομα (π.χ. /proc/1234 για μια διεργασία με PID 1234). Μέσα σε αυτόν τον φάκελο θα βρούμε αρχεία όπως:

* **status**: Περιέχει πληροφορίες για την κατάσταση της διεργασίας (όπως όνομα, κατάσταση, κατανάλωση μνήμης κ.λπ.).
* **cmdline**: Δείχνει την εντολή που εκτέλεσε τη διεργασία.
* **fd**: Περιέχει συμβολικούς συνδέσμους προς τα ανοιχτά αρχεία της διεργασίας.

Χρησιμοποιώντας την εντολή cat, μπορούμε να διαβάσουμε το περιεχόμενο αυτών των αρχείων και να εξετάσουμε τις πληροφορίες.

Ας δούμε ένα παράδειγμα με την αρχική διεργασία του συστήματος, η οποία έχει PID 1:

[user@host ~]$ sudo cat /proc/1/comm

systemd

[user@host ~]$ sudo head -5 /proc/1/status

Name: systemd

Umask: 0000

State: S (sleeping)

Tgid: 1

Ngid: 0

[user@host ~]$ sudo cat /proc/1/cmdline

/usr/lib/systemd/systemdrhgb--switched-root--system--deserialize31

Σε αυτό το παράδειγμα, βλέπουμε την εντολή **systemd** να εκτελείται με το PID 1, μαζί με πληροφορίες για την κατάσταση της διεργασίας και την ακριβή εντολή που εκτελέστηκε.

## Εκτέλεση και διαχείριση διεργασιών στο παρασκήνιο

Σε περιβάλλοντα Linux χωρίς γραφικό περιβάλλον, η αλληλεπίδραση του χρήστη με το σύστημα γίνεται μέσω του τερματικού. Η δυνατότητα εκτέλεσης διεργασιών στο παρασκήνιο (**background**) είναι κρίσιμη για την αποδοτική χρήση του συστήματος, επιτρέποντας την ταυτόχρονη εκτέλεση πολλαπλών εργασιών.

### Έναρξη διεργασίας στο παρασκήνιο

Για να εκτελέσουμε μια εντολή στο παρασκήνιο προσθέτουμε τον χαρακτήρα & στο τέλος της. Για παράδειγμα, η εντολή:

[user@host ~]$ find / -name \*.txt 1>textfiles 2>/dev/null &

[1] 14795

αναζητά όλα τα αρχεία με κατάληξη .txt σε ολόκληρο το σύστημα αρχείων. Ας αναλύσουμε την εντολή και την έξοδό της:

* find / -name "\*.txt": Η βασική εντολή αναζήτησης.
* 1> textfiles: Ανακατεύθυνση της τυπικής εξόδου (stdout) στο αρχείο textfiles.
* 2>/dev/null: Ανακατεύθυνση της τυπικής εξόδου σφαλμάτων (stderr) στο /dev/null, δηλαδή την απόρριψη των σφαλμάτων.
* &: Εκτέλεση της εντολής στο παρασκήνιο.

Η έξοδος της παραπάνω εντολής μας παρουσιάζει ότι [1] είναι ο αριθμός εργασίας (job number) και 14795 είναι το αναγνωριστικό διεργασίας (PID). Αυτό σημαίνει ότι η διεργασία ξεκίνησε στο παρασκήνιο και το τερματικό είναι πλέον διαθέσιμο για άλλες εντολές.

Σημαντικές παρατηρήσεις:

**Ανακατεύθυνση Εξόδου**: Είναι σημαντικό να ανακατευθύνουμε την τυπική έξοδο (stdout) και την τυπική έξοδο σφαλμάτων (stderr) σε αρχεία ή στο /dev/null όταν εκτελούμε διεργασίες στο παρασκήνιο, ώστε να μην παρεμβαίνουν στην αλληλεπίδραση μας με το τερματικό.

**Ενημέρωση Τερματισμού**: Όταν μια διεργασία στο παρασκήνιο ολοκληρωθεί, το τερματικό θα εμφανίσει ένα μήνυμα, όπως:

[1]+ Done find / -name "\*.txt" 1> textfiles 2>/dev/null

(ή Exit <κωδικός\_εξόδου> σε περίπτωση σφάλματος) ενημερώνοντάς μας για την ολοκλήρωση της διεργασίας.

### Διαχείριση διεργασιών στο παρασκήνιο με την εντολή jobs

Η εντολή jobs εμφανίζει τις διεργασίες που εκτελούνται στο παρασκήνιο στον τρέχοντα φλοιό (shell). Η εντολή jobs -l εμφανίζει επιπλέον το PID κάθε διεργασίας.

Παράδειγμα:

1. Ξεκινάμε μια διεργασία στο προσκήνιο, π.χ. sleep 40.
2. Πατώντας Ctrl+Z στέλνουμε το σήμα SIGTSTP στη διεργασία, το οποίο την σταματάει (Stopped) και επιστρέφει την προτροπή στο τερματικό.

[user@host ~]$ sleep 40

^Z

[2]+ Stopped sleep 40

Η εντολή jobs μας δείχνει ότι η εργασία sleep 40 σε κατάσταση Stopped. Το [2] είναι ο αριθμός εργασίας. Μάλιστα η διεργασία με το σύμβολο + είναι αυτή που θα επηρεαστεί από μια εντολή bg ή fg αν δεν δώσουμε αριθμό εργασίας.

### Μεταφορά διεργασιών μεταξύ προσκηνίου και παρασκηνίου

Η εντολή bg (background) μεταφέρει μια σταματημένη διεργασία στο παρασκήνιο. Για να μεταφέρουμε την sleep 40 στο παρασκήνιο, εκτελούμε:

bg %1

[1]+ sleep 40 &

jobs

[1]+ Running sleep 40 &

Το %1 αναφέρεται στην εργασία με αριθμό 1. Αν δεν δώσουμε αριθμό, η εντολή bg θα επηρεάσει την τελευταία σταματημένη διεργασία (αυτή που έχει το σύμβολο +).

Η εντολή fg (foreground) επαναφέρει μια διεργασία από το παρασκήνιο στο προσκήνιο. Για να επαναφέρουμε την sleep 40, εκτελούμε:

fg %1

ή γράφουμε απλά fg αν είναι η τελευταία σταματημένη ή τρέχουσα διεργασία στο παρασκήνιο.

### Αναφορά σε εργασίες

Εκτός από τους αριθμούς εργασίας, μπορούμε να αναφερθούμε σε μια εργασία με τους εξής συμβολισμούς:

* %string: Η τελευταία εργασία που το όνομά της ξεκινά με string.
* %?string: Η τελευταία εργασία που το όνομά της περιέχει το string.

Για παράδειγμα, αν έχουμε μια εργασία long\_process.sh, μπορούμε να την επαναφέρουμε στο προσκήνιο με fg %long.

## Σήματα διεργασιών

Σε ένα σύστημα Linux, εκτελούνται ταυτόχρονα εκατοντάδες διεργασίες, πολλές από τις οποίες δεν συνδέονται με τερματικά. Μάλιστα, σε πολλές περιπτώσεις, μπορεί να χρειαστεί να σταματήσουμε προσωρινά την εκτέλεση μιας διεργασίας, να την τερματίσουμε οριστικά. Όλες αυτές οι ενέργειες μπορούν να επιτευχθούν με την αποστολή του κατάλληλου σήματος διαδικασία που θα αναλύσουμε αναλυτικά παρακάτω.

### Αποστολή σημάτων με την εντολή kill

Στο Linux, τα σήματα αποτελούν έναν μηχανισμό ασύγχρονης επικοινωνίας μεταξύ διεργασιών. Αυτή η δυνατότητα είναι κρίσιμη σε περιβάλλοντα όπου πολλές διεργασίες εκτελούνται ταυτόχρονα, επιτρέποντας τον έλεγχο και τη διαχείρισή τους.

Η αποστολή ενός σήματος σε μια διεργασία πραγματοποιείται μέσω της εντολής kill. Η εντολή αυτή απαιτεί δύο παραμέτρους: το σήμα που θα σταλεί και το PID (Process ID) της διεργασίας-στόχου. Εάν δεν καθοριστεί κάποιο συγκεκριμένο σήμα, η kill αποστέλλει το προκαθορισμένο σήμα τερματισμού, το SIGTERM.

Για παράδειγμα, με την εντολή ps μπορούμε να δούμε τις διεργασίες που εκτελούνται σε δύο τερματικά:

[user@host ~]$ ps -t pts/0,pts/1

PID TTY TIME CMD

2931 pts/0 00:00:00 bash

3155 pts/1 00:00:00 bash

3300 pts/1 00:00:00 sleep

3301 pts/0 00:00:00 ps

Στο παραπάνω παράδειγμα, παρατηρούμε ότι η διεργασία sleep εκτελείται στο τερματικό pts/1. Για να τη σταματήσουμε από το τερματικό pts/0, χρησιμοποιούμε την εντολή:

[user@host ~]$ kill 3300

Εκτελώντας ξανά την εντολή ps, επιβεβαιώνουμε ότι η διεργασία sleep έχει τερματιστεί:

[user@host ~]$ ps -t pts/0,pts/1

PID TTY TIME CMD

2931 pts/0 00:00:00 bash

3155 pts/1 00:00:00 bash

3315 pts/0 00:00:00 ps

### Βασικά Σήματα στο Linux

Υπάρχουν πολλά σήματα που μπορούν να σταλούν σε διεργασίες, καθένα με συγκεκριμένο σκοπό. Μερικά από τα βασικά σήματα παρουσιάζονται στον παρακάτω πίνακα:

|  |  |  |
| --- | --- | --- |
| **Κωδικός** | **Σήμα** | **Περιγραφή** |
| 2 | **SIGINT** | Τερματισμός διεργασίας. Μπορεί να αγνοηθεί ή να «συλληφθεί» για εκτέλεση συγκεκριμένων ενεργειών (Ctrl+C). |
| 9 | **SIGKILL** | Άμεσος τερματισμός διεργασίας. Δεν μπορεί να αγνοηθεί ή να «συλληφθεί». |
| 15 | **SIGTERM** | Τυπικός τερματισμός διεργασίας. Μπορεί να αγνοηθεί ή να «συλληφθεί» για προσαρμοσμένη διαχείριση. |
| 18 | **SIGCONT** | Επαναφορά διεργασίας που έχει σταματήσει προσωρινά (με SIGSTOP). |
| 19 | **SIGSTOP** | Προσωρινή παύση διεργασίας. Μπορεί να ενεργοποιηθεί με Ctrl+Z. |

Θα μπορούσαμε για παράδειγμα να στείλουμε το σήμα SIGSTOP (19) στη διεργασία sleep για προσωρινή παύση της:

[user@host ~]$ ps -t pts/0,pts/1 -o pid,comm,tty,state

PID COMMAND TT S

2931 bash pts/0 S

3155 bash pts/1 S

3453 sleep pts/1 S

3454 ps pts/0 R

[user@host ~]$ kill -SIGSTOP 3453

[user@host ~]$ ps -t pts/0,pts/1 -o pid,comm,tty,state

PID COMMAND TT S

2931 bash pts/0 S

3155 bash pts/1 S

3453 sleep pts/1 T

3468 ps pts/0 R

Η κατάσταση T υποδεικνύει ότι η διεργασία έχει παύσει προσωρινά. Για να συνεχίσει η εκτέλεση, αποστέλλουμε το σήμα SIGCONT (18):

[user@host ~]$ kill -SIGCONT 3453

[user@host ~]$ ps -t pts/0,pts/1 -o pid,comm,tty,state

PID COMMAND TT S

2931 bash pts/0 S

3155 bash pts/1 S

3453 sleep pts/1 S

3490 ps pts/0 R

### Αποστολή σημάτων με την εντολή killall

Για πιο γρήγορη διαχείριση, όταν γνωρίζουμε το όνομα της εντολής, μπορούμε να χρησιμοποιήσουμε την εντολή killall. Η killall τερματίζει όλες τις διεργασίες που εκτελούν μια συγκεκριμένη εντολή, χωρίς να απαιτείται το PID. Για παράδειγμα, για να τερματίσουμε το script zombie.sh που εκτελεί ο χρήστης user, χρησιμοποιούμε:

[user@host ~]$ killall -u user zombie.sh

Μπορούμε να δούμε τον πλήρη κατάλογο των σημάτων που υποστηρίζονται στο σύστημα μας μέσω της εντολής man 7 signal

## Προτεραιότητες διεργασιών

Ένα από τα βασικά χαρακτηριστικά που λαμβάνει υπόψη ο πυρήνας του Linux για να επιλέξει ποια διεργασία θα εκτελεστεί από τον επεξεργαστή είναι η προτεραιότητά της. Κάθε διεργασία διαθέτει έναν αριθμό προτεραιότητας (**niceness**) που κυμαίνεται από -20 έως 19, όπου μικρότερες τιμές αντιστοιχούν σε υψηλότερη προτεραιότητα. Ο προκαθορισμένος αριθμός προτεραιότητας για μια διεργασία είναι το 0, ενώ οι διεργασίες που εκτελούνται από κανονικούς χρήστες μπορούν να έχουν προτεραιότητα μεγαλύτερη ή ίση με 0. Αυτό σημαίνει ότι, υπό κανονικές συνθήκες, οι διεργασίες ενός χρήστη εκτελούνται με τη μέγιστη δυνατή προτεραιότητα που επιτρέπεται για εκείνον.

Εάν ένας χρήστης θέλει να εκτελέσει μια διεργασία με χαμηλότερη προτεραιότητα, μπορεί να το κάνει χρησιμοποιώντας την εντολή nice. Για παράδειγμα, η παρακάτω εντολή εκτελεί το script myscript.sh με αριθμό προτεραιότητας 10:

[user@host ~]$ nice -n 10 ./myscript.sh

Για να εκτελεστεί μια διεργασία με υψηλότερη προτεραιότητα (αρνητική τιμή προτεραιότητας), απαιτούνται δικαιώματα διαχειριστή (root). Σε περίπτωση που ο χρήστης επιχειρήσει να το κάνει χωρίς τα κατάλληλα δικαιώματα, το σύστημα θα εμφανίσει μήνυμα σφάλματος:

[user@host ~]$ nice -n -10 ./myscript.sh

nice: cannot set niceness: Permission denied

[user@host ~]$ sudo nice -n -10 ./myscript.sh

### Αλλαγή Προτεραιότητας σε Εκτελούμενη Διεργασία

Η προτεραιότητα μιας διεργασίας που εκτελείται ήδη μπορεί να τροποποιηθεί με τη χρήση της εντολής renice. Ωστόσο, όπως και με την nice, οι κανονικοί χρήστες μπορούν να μειώσουν μόνο την προτεραιότητα της διεργασίας (να αυξήσουν την τιμή niceness). Για να αυξήσουμε την προτεραιότητα (μειώνοντας την τιμή niceness) μιας κρίσιμης διεργασίας με PID 1234, απαιτούνται δικαιώματα root, και η εντολή εκτελείται ως εξής:

[user@host ~]$ sudo renice -5 -p 1234

## Παρακολούθηση και διαχείριση διεργασιών με την χρήση εργαλείων

Το Linux διαθέτει ισχυρά εργαλεία για την παρακολούθηση και διαχείριση των διεργασιών του συστήματος, τόσο σε περιβάλλον γραμμής εντολών όσο και σε γραφικό περιβάλλον. Η εντολή **top** αποτελεί ένα ολοκληρωμένο εργαλείο που παρέχει λεπτομερή εποπτεία και έλεγχο των τρεχουσών διεργασιών σε πραγματικό χρόνο μέσω του τερματικού. Παράλληλα, το GNOME **System Monitor** προσφέρει τις ίδιες δυνατότητες διαχείρισης μέσω μιας εύχρηστης γραφικής διεπαφής.

### Παρακολούθηση και διαχείριση διεργασιών με την εντολή top

Η εντολή top παρέχει αναλυτικές πληροφορίες για τις διεργασίες που εκτελούνται στο σύστημα. Η έξοδος της εντολής ανανεώνεται δυναμικά κάθε τρία δευτερόλεπτα, παρουσιάζοντας την τρέχουσα κατάσταση των διεργασιών στο σύστημα. Η εκτέλεση της top συνεχίζεται μέχρι να πατήσουμε το πλήκτρο q, το οποίο τη σταματά.

Στο πάνω μέρος της οθόνης εμφανίζονται γενικές πληροφορίες για τη λειτουργία του συστήματος, όπως:

* Ο χρόνος λειτουργίας του συστήματος.
* Ο αριθμός χρηστών που είναι συνδεδεμένοι.
* Ο συνολικός αριθμός διεργασιών.
* Η χρήση του επεξεργαστή και της μνήμης.

A screenshot of a computer

Description automatically generated

Εικόνα 1 Η έξοδος της εντολής top

Αμέσως παρακάτω, εμφανίζεται μια λίστα με τις διεργασίες του συστήματος, όπου κάθε γραμμή αντιστοιχεί σε μία διεργασία. Για κάθε διεργασία παρουσιάζονται διάφορα πεδία, όπως το PID, η χρήση της CPU (%CPU), η χρήση της μνήμης (%MEM) και άλλες ιδιότητες.

**Ταξινόμηση και προσαρμογή των πεδίων**

Ο προκαθορισμένος τρόπος ταξινόμησης των διεργασιών είναι το ποσοστό χρήσης της CPU. Πατώντας το πλήκτρο f, εμφανίζεται ένα μενού που επιτρέπει την επιλογή των πεδίων που θέλουμε να προβάλουμε, καθώς και την επιλογή του πεδίου που θα χρησιμοποιηθεί για την ταξινόμηση. Μπορούμε να αντιστρέψουμε τη σειρά ταξινόμησης πατώντας το πλήκτρο R.

A screenshot of a computer screen

Description automatically generated

Εικόνα 2 Οι επιλογές της βοήθειας της εντολής top

**Διαχείριση των διεργασιών με την top**

Η top δεν περιορίζεται μόνο στην παρακολούθηση των διεργασιών. Παρέχει επίσης εργαλεία για τη διαχείρισή τους. Πατώντας το πλήκτρο h, εμφανίζεται η βοήθεια της εντολής με μια λίστα ενεργειών που μπορούν να εκτελεστούν μέσα από την top.

Ένα σημαντικό χαρακτηριστικό είναι η δυνατότητα αποστολής σημάτων σε διεργασίες. Πατώντας το πλήκτρο k, η top μας ζητά να εισάγουμε το **PID** της διεργασίας στην οποία θέλουμε να στείλουμε ένα σήμα. Αφού εισάγουμε το **PID**, μας ζητείται να καθορίσουμε τον αριθμό του σήματος που θα σταλεί (όπως τα γνωστά σήματα **SIGTERM**, **SIGKILL** κ.λπ.).

### Παρακολούθηση και διαχείριση διεργασιών με το System Monitor

Το λογισμικό **System Monitor**, που εγκαθίσταται μαζί με το γραφικό περιβάλλον **GNOME**, προσφέρει δυνατότητες παρακολούθησης της λειτουργίας του συστήματος και διαχείρισης των διεργασιών που εκτελούνται σε αυτό. Η διεπαφή του αποτελείται από τρεις βασικές καρτέλες, εκ των οποίων η καρτέλα **Resources** παρέχει πληροφορίες για τον φόρτο του συστήματος, όπως η χρήση του επεξεργαστή, της μνήμης και του δικτύου, όπως φαίνεται στην παρακάτω εικόνα.

A screenshot of a computer

Description automatically generated

Εικόνα 3 Η καρτέλα Resources

Η καρτέλα **Processes** εμφανίζει τη λίστα των διεργασιών που εκτελούνται στο σύστημα, μαζί με βασικές πληροφορίες, όπως ο χρήστης στον οποίο ανήκει η διεργασία, το **PID**, το ποσοστό χρήσης του επεξεργαστή και άλλες λεπτομέρειες. Οι διεργασίες μπορούν να ταξινομηθούν κάνοντας κλικ στον τίτλο της αντίστοιχης στήλης. Επιπλέον, οι ρυθμίσεις εμφάνισης της λίστας των διεργασιών διαχειρίζονται από το κουμπί επιλογών που βρίσκεται στο πάνω δεξί μέρος του παραθύρου, όπως φαίνεται στην εικόνα.

A screenshot of a computer

Description automatically generated

Εικόνα 4 Ρυθμίσεις εμφάνισης των διεργασιών

Μερικές από τις βασικές ρυθμίσεις που προσφέρει το εργαλείο περιλαμβάνουν:

* Την επιλογή εμφάνισης μόνο των ενεργών διεργασιών, όλων των διεργασιών ή μόνο αυτών που ανήκουν στον χρήστη.
* Την εμφάνιση της ιεραρχίας εξάρτησης μεταξύ διεργασιών (σχέσεις πατρικής-απογόνου).
* Την προσαρμογή των πεδίων που εμφανίζονται για κάθε διεργασία μέσω της επιλογής Preferences.

Εκτός από την παρακολούθηση, το System Monitor προσφέρει και λειτουργίες διαχείρισης διεργασιών.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, οθόνη, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα 5 Διαχείριση διεργασιών μέσα από το System Monitor

Όταν κάνουμε δεξί κλικ σε μια διεργασία, εμφανίζεται ένα μενού ενεργειών διαχείρισης. Από το μενού αυτό, μπορούμε να:

* Δούμε αναλυτικές πληροφορίες για τη διεργασία.
* Προβάλουμε τα αρχεία που έχει «ανοίξει» η διεργασία (file descriptors).
* Αλλάξουμε την προτεραιότητα της διεργασίας.
* Στείλουμε βασικά σήματα στη διεργασία, όπως Stop, Continue, End και Kill.

## Άσκηση – Διαχείριση διεργασιών από την γραμμή εντολών

### Εργασία 1. Εμφάνιση πληροφοριών για τις διεργασίες

**Σενάριο 1: Εμφάνιση αναλυτικών πληροφοριών για όλες τις διεργασίες που εκτελούνται στο σύστημα.**

Στόχος: Να παρουσιάσουμε όλες τις διεργασίες που εκτελούνται στο σύστημα από όλους τους χρήστες, παρέχοντας λεπτομερείς πληροφορίες.

Η εντολή ps εμφανίζει τις διεργασίες του συστήματος. Από προεπιλογή, όμως, δείχνει μόνο τις διεργασίες που εκτελεί ο τρέχων χρήστης στη συγκεκριμένη σύνοδο. Για να εμφανίσουμε όλες τις διεργασίες ανεξαρτήτως χρήστη και συνόδου, χρησιμοποιούμε την επιλογή -e. Για αναλυτική παρουσίαση των πληροφοριών, προσθέτουμε την επιλογή -f.

[user@host ~]$ ps -ef | head -5

UID PID PPID C STIME TTY TIME CMD

root 1 0 0 08:44 ? 00:00:01 /usr/lib/systemd/systemd rhgb --switched-root --system --deserialize 31

root 2 0 0 08:44 ? 00:00:00 [kthreadd]

root 3 2 0 08:44 ? 00:00:00 [rcu\_gp]

root 4 2 0 08:44 ? 00:00:00 [rcu\_par\_gp]

**Σενάριο 2: Εμφάνιση διεργασιών ενός συγκεκριμένου χρήστη**

Στόχος: Να εμφανίσουμε τις διεργασίες που εκτελεί ένας συγκεκριμένος χρήστης (π.χ., root) με αναλυτική παρουσίαση.

Χρησιμοποιούμε την επιλογή -u <όνομα\_χρήστη> για να φιλτράρουμε τις διεργασίες ενός χρήστη. Για αναλυτική εμφάνιση, συνδυάζουμε την επιλογή -f.

[user@host ~]$ ps -u root -f | tail -5

root 3314 2 0 09:33 ? 00:00:00 [kworker/1:1-events]

root 3316 2 0 09:35 ? 00:00:00 [kworker/1:0H-ttm]

root 3317 2 0 09:35 ? 00:00:00 [kworker/1:1H-ttm]

root 3318 2 0 09:35 ? 00:00:00 [kworker/1:2H-kblockd]

root 3319 2 0 09:35 ? 00:00:00 [kworker/1:3H]

**Σενάριο 3: Λίστα με όλες τις πληροφορίες που παρέχει η ps**

Στόχος: Να δείξετε όλες τις διαθέσιμες πληροφορίες που μπορεί να επιστρέψει η ps.

Για να εμφανίσουμε τη λίστα με όλα τα πεδία που μπορεί να εμφανίσει η ps, χρησιμοποιούμε την επιλογή L (χωρίς παύλα).

[user@host ~]$ ps L | tail -5

vsize VSZ

vsz VSZ

wchan WCHAN

wname WCHAN

zone ZONE

**Σενάριο 4: Παρουσίαση συγκεκριμένων πληροφοριών για όλες τις διεργασίες**

Στόχος: Να εμφανίσετε πληροφορίες όπως το όνομα χρήστη, το ποσοστό χρήσης CPU, το PID και την εντολή κάθε διεργασίας.

Η επιλογή -o επιτρέπει την προσαρμογή της εξόδου καθορίζοντας τα πεδία που θέλουμε να εμφανίσουμε. Για όλες τις διεργασίες, χρησιμοποιούμε και την επιλογή -e.

[user@host ~]$ ps -e -o ruser,%cpu,pid,comm |head -5

RUSER %CPU PID COMMAND

root 0.0 1 systemd

root 0.0 2 kthreadd

root 0.0 3 rcu\_gp

root 0.0 4 rcu\_par\_gp

**Σενάριο 5: Ταξινόμηση των αποτελεσμάτων της ps**

Στόχος: Να ταξινομήσουμε τις πληροφορίες από το Σενάριο 4 με βάση το όνομα της εντολής, σε αντίστροφη σειρά.

Για ταξινόμηση, χρησιμοποιούμε την επιλογή --sort=<πεδίο>. Αν θέλουμε αντίστροφη σειρά, προσθέτουμε το σύμβολο - πριν από το όνομα του πεδίου.

[user@host ~]$ ps -e -o ruser,%cpu,pid,comm --sort=comm | head -5

RUSER %CPU PID COMMAND

user 0.0 1992 (sd-pam)

root 0.0 948 ModemManager

root 0.0 1066 NetworkManager

root 0.0 908 VGAuthService

**Σενάριο 6: Εμφάνιση εντολής δημιουργίας διεργασίας με PID 1**

Στόχος: Να εμφανίσουμε την εντολή που δημιούργησε τη διεργασία με PID 1, μαζί με τις επιλογές της.

Χρησιμοποιούμε την επιλογή -p <pidlist> για να φιλτράρουμε με βάση το PID. Το πεδίο args επιστρέφει την πλήρη εντολή με τις παραμέτρους της.

[user@host ~]$ ps -p 1 -o args

COMMAND

/usr/lib/systemd/systemd rhgb --switched-root --system --deserialize 3

**Σενάριο 7: Εύρεση πληροφοριών μέσα από το εικονικό σύστημα αρχείων /proc**

Στόχος: Να βρούμε την εντολή που δημιούργησε τη διεργασία με PID 1 μέσω του καταλόγου /proc.

Ο κατάλογος /proc/<pid> περιέχει πληροφορίες για την διεργασία με PID <pid>. Το αρχείο comm που υπάρχει μέσα σε αυτόν περιέχει το όνομα της εντολής.

[user@host ~]$ cat /proc/1/comm

systemd

### Εργασία 2. Αποστολή σημάτων σε διεργασίες

**Σενάριο 1: Τερματισμός διεργασίας που εκτελείται στο παρασκήνιο**

Δημιουργία διεργασίας στο παρασκήνιο:

Χρησιμοποιούμε την εντολή sleep για να δημιουργήσουμε μια διεργασία που εκτελείται στο παρασκήνιο. Αποθηκεύουμε το PID της στη μεταβλητή MYPROC2 και εμφανίζουμε το PID στην οθόνη:

[user@host ~]$ (sleep 3600 & MYPROC2=$! && echo PID is: $MYPROC2) 2>/dev/null

Προβολή πληροφοριών της διεργασίας:

Εμφανίζουμε λεπτομερείς πληροφορίες για τη διεργασία χρησιμοποιώντας την εντολή ps. Στη θέση του PID (3555 στο παράδειγμα) εισάγουμε το PID που εμφανίστηκε από την προηγούμενη εντολή.

[user@host ~]$ ps -f -p 3555

UID PID PPID C STIME TTY TIME CMD

user 3555 2011 0 10:32 pts/1 00:00:00 sleep 3600

Αποστολή σήματος τερματισμού (SIGTERM):

Τερματίζουμε τη διεργασία με την εντολή kill και επιβεβαιώνουμε ότι έχει τερματιστεί.

[user@host ~]$ kill -SIGTERM 3555

Αν εκτελέσουμε την προηγούμενη εντολή ps δεν θα πρέπει να μας εμφανίζει τίποτα.

[user@host ~]$ ps -f -p 3555

UID PID PPID C STIME TTY TIME CMD

**Σενάριο 2: Τερματισμός διεργασίας που αγνοεί το σήμα SIGINT**

**Δημιουργία διεργασίας που παγιδεύει το SIGINT:**

Δημιουργούμε ένα script Bash (loop.sh) που εκτελεί μια ατέρμονη διαδικασία και παγιδεύει το σήμα SIGINT. Αποθηκεύουμε τον κώδικα και τον κάνουμε εκτελέσιμο. Εκτελούμε το script.

#!/bin/bash

trap "echo 'Ignoring Ctrl+C. The process will continue...'" SIGINT

function endless\_process {

while true; do

echo "Running... PID: $$"

sleep 2

done

}

endless\_process

[user@host ~]$ chmod +x loop.sh

[user@host ~]$ ./loop.shss\_process

**Παρατήρηση συμπεριφοράς διεργασίας στο σήμα SININT (Ctrl+C):**

Αρχικά παρατηρούμε ότι η διεργασία εκτελείται συνέχεια και «καταλαμβάνει» το τερματικό. Προσπαθούμε να την τερματίσουμε αποστέλλοντας της το σήμα SIGINT με την χρήση του συνδυασμού πλήκτρων Ctrl-C. Διαπιστώνουμε ότι το σήμα SIGINT αγνοείται και η διεργασία συνεχίζει να εκτελείται.

**Τερματισμός με αποστολή σήματος SIGKILL από άλλο τερματικό:**

Σε διαφορετικό τερματικό, βρίσκουμε το PID της διεργασίας και στέλνουμε σήμα SIGKILL, που τερματίζει άμεσα τη διεργασία. Αυτό γίνεται γιατί το σήμα SIGKILL δεν μπορεί να «παγιδευτεί» από μία διεργασία.

[user@host ~]$ kill -SIGKILL 3608

**Σενάριο 3: Μελέτη του σήματος SIGHUP**

**Δημιουργία διεργασίας στο παρασκήνιο:**

Εκκινούμε την εντολή sleep στο παρασκήνιο.

[user@host ~]$ sleep 4000&

**Προβολή πληροφοριών της διεργασίας:**

Σε διαφορετικό τερματικό, εμφανίζουμε πληροφορίες για τη διεργασία.

[user@host ~]$ ps -ef | grep -v grep | grep sleep

user 3675 3177 0 10:57 pts/1 00:00:00 sleep 4000

**Τερματισμός της διεργασίας από σήμα SIGHUP:**

Κλείνουμε το τερματικό από όπου ξεκίνησε η διεργασία. Παρατηρούμε ότι η διεργασία sleep τερματίζεται λόγω του σήματος SIGHUP που αποστέλλεται αυτόματα από το τερματικό σε όλες τις διεργασίες του όταν τερματίζεται.

**Αποφυγή τερματισμού με χρήση της nohup:**

Ξεκινάμε την εντολή sleep με την εντολή nohup, ώστε η διεργασία να συνεχίσει να εκτελείται ακόμη και μετά το κλείσιμο του τερματικού.

[user@host ~]$ nohup sleep 50000 &

**Επιβεβαίωση της εκτέλεσης:**

Μετά το κλείσιμο του τερματικού, χρησιμοποιούμε την εντολή ps για να επιβεβαιώσουμε ότι η διεργασία συνεχίζει να εκτελείται.

### Εργασία 3. Παρακολούθηση της λειτουργίας του συστήματος με την χρήση της top

**Σενάριο 1: Διαμόρφωση της παρουσίασης της top**

Η εντολή top είναι ένα ισχυρό εργαλείο για την παρακολούθηση διεργασιών σε πραγματικό χρόνο. Στο παρακάτω σενάριο, θα εξετάσουμε πώς να προσαρμόσουμε την προβολή και να φιλτράρουμε τις πληροφορίες:

**Εμφάνιση διεργασιών με τη μεγαλύτερη κατανάλωση μνήμης:**

Πατάμε το πλήκτρο **M** για ταξινόμηση των διεργασιών με βάση τη χρήση μνήμης σε φθίνουσα σειρά.

Για να επιστρέψουμε στην προηγούμενη ταξινόμηση (π.χ., με βάση τη χρήση CPU), πατάμε **P**.

**Εμφάνιση διεργασιών που εκτελούνται από συγκεκριμένο χρήστη:**

Πατάμε το πλήκτρο **u** και εισάγουμε το όνομα χρήστη που μας ενδιαφέρει. Η προβολή περιορίζεται στις διεργασίες που ανήκουν σε αυτόν τον χρήστη.

**Προβολή χρόνου λειτουργίας του συστήματος:**

Στην κορυφή της οθόνης της top, παρατηρούμε τον χρόνο λειτουργίας του συστήματος (uptime), που εμφανίζεται μαζί με άλλες γενικές πληροφορίες.

**Ταξινόμηση διεργασιών με βάση το όνομα της εντολής (comm):**

Πατάμε **F** για να ανοίξουμε το μενού επιλογών ταξινόμησης.

Επιλέγουμε την επιλογή **comm** (Command) πατώντας το αντίστοιχο πλήκτρο.

Στη συνέχεια, πατάμε **s** για ενεργοποίηση της ταξινόμησης και **q** για επιστροφή στην κανονική προβολή.

**Αντιστροφή της τρέχουσας ταξινόμησης:**

Πατάμε το πλήκτρο **R** για αντιστροφή της κατεύθυνσης ταξινόμησης (αύξουσα ή φθίνουσα).

**Σενάριο 2: Εντοπισμός και τερματισμός διεργασίας με υψηλή χρήση CPU**

Στο συγκεκριμένο σενάριο, θα εντοπίσουμε και θα τερματίσουμε μια διεργασία που καταναλώνει μεγάλο ποσοστό της CPU:

**Δημιουργία αρχείου μεγάλου μεγέθους:**

Εκτελούμε την παρακάτω εντολή για να δημιουργήσουμε ένα μεγάλο αρχείο (βεβαιωνόμαστε ότι υπάρχει αρκετός ελεύθερος χώρος στον δίσκο):

[user@host ~]$ fallocate -l 10G large.dat

**Εκκίνηση της top:**

Σε ένα άλλο τερματικό, εκτελούμε την εντολή top και φροντίζουμε οι διεργασίες να είναι ταξινομημένες με βάση τη χρήση της CPU (σε φθίνουσα σειρά). Αυτό μπορεί να γίνει πατώντας το πλήκτρο **P**.

**Εκτέλεση διεργασίας υψηλής κατανάλωσης CPU:**

Σε ένα τρίτο τερματικό, ξεκινάμε τη συμπίεση του μεγάλου αρχείου που δημιουργήσαμε με την παρακάτω εντολή:

[user@host ~]$ tar -czf archive.tar.gz large.dat

**Παρακολούθηση της διεργασίας στη top:**

Στην οθόνη της top, παρατηρούμε ότι οι διεργασίες gzip και tar εμφανίζονται στις πρώτες θέσεις, καθώς καταναλώνουν το μεγαλύτερο ποσοστό της CPU.

**Τερματισμός διεργασίας:**

Για να τερματίσουμε τη διεργασία, ακολουθούμε τα εξής βήματα:

* Πατάμε το πλήκτρο **k** (kill).
* Εισάγουμε το **PID** της διεργασίας tar που εμφανίζεται στην top.
* Επιλέγουμε το σήμα SIGKILL (9) για να τερματίσουμε άμεσα τη διεργασία.

**Επιβεβαίωση τερματισμού:**

Επιστρέφουμε στο τερματικό όπου εκτελείται η εντολή tar και παρατηρούμε ότι η διεργασία έχει τερματιστεί.

### Εργασία 4: Αλλαγή της προτεραιότητας εκτέλεσης μιας διεργασίας με nice/renice

**Σενάριο 1: Εμφάνιση προτεραιότητας εκτέλεσης μιας διεργασίας**

Στο σενάριο αυτό, θα δούμε πώς να εντοπίσουμε την προτεραιότητα εκτέλεσης μιας διεργασίας που εκτελεί έντονες υπολογιστικές εργασίες.

**Εκκίνηση διεργασίας έντονης χρήσης CPU:**

Εκτελούμε την παρακάτω εντολή, η οποία υπολογίζει τις κατακερματιστικές τιμές (hash) όλων των αρχείων του συστήματος, παραλείποντας τον κατάλογο /proc:

find / -path '/proc/\*' -prune -o -type f -exec md5sum {} \; > /dev/null

**Εντοπισμός της διεργασίας (PID):**

Σε ένα δεύτερο τερματικό, εντοπίζουμε το PID της διεργασίας find με την ακόλουθη εντολή:

[user@host ~]$ ps -e | grep find

**Προβολή προτεραιότητας διεργασίας:**

Εκτελούμε την εντολή ps με την επιλογή -l για να δούμε αναλυτικές πληροφορίες, συμπεριλαμβανομένης της τρέχουσας προτεραιότητας (παράμετρος PRI) και της τιμής nice (παράμετρος NI):

[user@host ~]$ ps -l <PID>

**Τερματισμός διεργασίας:**

Αν η εντολή find δεν έχει ολοκληρωθεί, τερματίζουμε τη διεργασία είτε πατώντας Ctrl+C στο τερματικό της είτε με την εντολή kill από ένα άλλο τερματικό.

**Σενάριο 2: Εκτέλεση διεργασίας με διαφορετικές προτεραιότητες**

Σε αυτό το σενάριο, θα εξετάσουμε πώς να εκτελέσουμε τη διεργασία find με διαφορετικές τιμές nice και πώς να τροποποιήσουμε την προτεραιότητά της κατά τη διάρκεια της εκτέλεσης.

**Εκτέλεση διεργασίας με χαμηλή προτεραιότητα (nice +19):**

Η προτεραιότητα των διεργασιών στο Linux κυμαίνεται από -20 (υψηλότερη προτεραιότητα) έως +19 (χαμηλότερη). Εκτελούμε τη διεργασία find με χαμηλή προτεραιότητα:

[user@host ~]$ nice -n 19 find / -path '/proc/\*' -prune -o -type f -exec md5sum {} \; > /dev/null

**Εντοπισμός και επαλήθευση προτεραιότητας:**

Επαναλαμβάνουμε τα βήματα του Σενάριου 1 για να εντοπίσουμε το PID της διεργασίας και να επαληθεύσουμε την τιμή nice και την προτεραιότητά της.

**Τερματισμός διεργασίας:**

Όταν ολοκληρώσουμε την παρατήρηση, τερματίζουμε τη διεργασία χρησιμοποιώντας τις εντολές που αναφέρθηκαν στο προηγούμενο σενάριο.

**Εκτέλεση διεργασίας με υψηλή προτεραιότητα (nice -20):**

Χρησιμοποιούμε τη δυνατότητα sudo για να εκτελέσουμε τη διεργασία με την υψηλότερη δυνατή προτεραιότητα:

[user@host ~]$ sudo nice -n -20 find / -path '/proc/\*' -prune -o -type f -exec md5sum {} \; > /dev/null

**Μεταβολή προτεραιότητας κατά την εκτέλεση:**

Πατάμε Ctrl+Z για να σταματήσουμε προσωρινά τη διεργασία στο τερματικό.

Μεταβάλλουμε την προτεραιότητα της διεργασίας (π.χ., σε -10) χρησιμοποιώντας την εντολή renice:

[user@host ~]$ sudo renice -n -10 <PID>

Συνεχίζουμε την εκτέλεση της διεργασίας στο τερματικό όπου διακόπηκε, πληκτρολογώντας:

[user@host ~]$ fg

**Επαλήθευση αλλαγής προτεραιότητας:**

Επαναλαμβάνουμε τη διαδικασία προβολής πληροφοριών με την εντολή ps για να επιβεβαιώσουμε τη νέα τιμή nice και την προτεραιότητα.

## Ερωτήσεις αυτοαξιολόγησης

1. Με ποια εντολή μπορούμε να εκτελέσουμε με χαμηλότερη προτεραιότητα από την προκαθορισμένη;

Α. low

Β. slow

Γ. nice

Δ. priority

1. Ποια από τις παρακάτω επιλογές της ps εμφανίζει όλες τις διεργασίες ανεξάρτητα από τον χρήστη ή το τερματικό;

Α. -o

Β. -f

Γ. -a

Δ. -e

1. Ποιος είναι ο σκοπός των σημάτων (signals) στο Linux;

Α. Να αποθηκεύουν δεδομένα για τις διεργασίες

Β. Να ελέγχουν την πρόσβαση των διεργασιών στους πόρους του συστήματος

Γ. Να παρέχουν έναν μηχανισμό ασύγχρονης επικοινωνίας μεταξύ των διεργασιών

Δ. Να ορίζουν την ιεραρχία των διεργασιών

1. Ποια από τις παρακάτω προτάσεις είναι σωστή σχετικά με μία “parent” και μία “child” διεργασίας;

Α. Δεν υπάρχει καμία σχέση μεταξύ τους

Β. Η “parent” ελέγχει την πρόσβαση της “child” στους πόρους του συστήματος

Γ. Κάθε “child” δημιουργείται από μία “parent”

Δ. Οι “parent” έχουν μεγαλύτερη προτεραιότητα από τις “child”

1. Ποια από τις παρακάτω περιπτώσεις οδηγεί στη δημιουργία μιας νέας "συνόδου" (session) στο Linux;

Α. Όταν εκτελείται μια εντολή στο παρασκήνιο

Β. Όταν ένας χρήστης συνδέεται στο σύστημα μέσω ενός διαδραστικού φλοιού

Γ. Όταν μια διεργασία τερματίζεται

Δ. Όταν μια διεργασία αλλάζει προτεραιότητα

1. Με ποιον τρόπο εκτελούμε μια διεργασία στο παρασκήνιο;

Α. Προσθέτουμε τον χαρακτήρα & στην αρχή της εντολής

Β. Προσθέτουμε τον χαρακτήρα & στο τέλος της εντολής

Γ. Με την εντολή nice

Δ. Εκτελούμε την διεργασία σε άλλο τερματικό

1. Ποια από τις παρακάτω επιλογές της εντολής ps χρησιμοποιείται για να καθορίσουμε τη μορφή της εξόδου, επιλέγοντας συγκεκριμένα πεδία που θέλουμε να εμφανιστούν;

Α. -o

Β. -f

Γ. -p

Δ. -e

1. Ποια από τις παρακάτω προτάσεις περιγράφει σωστά τη λειτουργία της εντολής top στο Linux;

Α. Εμφανίζει στατικές πληροφορίες για τις διεργασίες του συστήματος

Β. Παρακολουθεί και εμφανίζει δυναμικά πληροφορίες για τις διεργασίες του συστήματος σε πραγματικό χρόνο

Γ. Χρησιμοποιείται για την εκκίνηση διεργασιών στο παρασκήνιο

Δ. Εμφανίζει μόνο τις διεργασίες που εκτελούνται από τον χρήστη root

1. Με ποια εντολή μπορώ να στείλω ένα σήμα σε μία διεργασία;

Α. kill

Β. signal

Γ. terminate

Δ. nice

1. Ποια είναι η μεγαλύτερη προτεραιότητα με την οποία μπορεί να εκτελέσει ένας κανονικός χρήστης μία διεργασία;

Α. 19

Β.-19

Γ.20

Δ.0

### Απαντήσεις στις ερωτήσεις αυτοαξιολόγησης

* + - 1. Γ
      2. Δ
      3. Γ
      4. Γ
      5. Β
      6. Β
      7. Α
      8. Β
      9. Α
      10. Δ

# Διαχείριση Υπηρεσιών και Δαιμόνων

Οι υπηρεσίες (services) και οι δαίμονες (daemons) στο Linux είναι βασικές συνιστώσες του λειτουργικού συστήματος που επιτρέπουν την εκτέλεση διαφόρων λειτουργιών και προγραμμάτων παρασκηνίου χωρίς την αλληλεπίδραση του χρήστη. Οι υπηρεσίες είναι προγράμματα που εκτελούνται συνεχώς στο παρασκήνιο και είναι υπεύθυνα για τη διαχείριση πόρων του συστήματος και την παροχή λειτουργικότητας, όπως δικτύωση, πρόσβαση σε αρχεία, ασφάλεια και παρακολούθηση. Οι δαίμονες είναι ένα είδος υπηρεσίας, αλλά δεν περιορίζονται μόνο σε βασικές λειτουργίες του συστήματος και μπορούν να εξυπηρετούν πιο εξειδικευμένους σκοπούς.

## Υπηρεσίες (Services)

Οι υπηρεσίες είναι λογισμικά ή διεργασίες που εκτελούνται στο παρασκήνιο για να εξυπηρετούν συγκεκριμένες ανάγκες και λειτουργίες του συστήματος, όπως το ξεκίνημα ή το σταμάτημα του δικτύου, η καταγραφή των συμβάντων, και η διαχείριση πόρων.

#### Τα κύρια χαρακτηριστικά συνοψίζονται παρακάτω:

|  |  |
| --- | --- |
| **Παραμένουν σε λειτουργία** | Συνήθως, συνεχίζουν να τρέχουν στο παρασκήνιο και είναι ενεργές μέχρι να σταματήσουν χειροκίνητα ή να γίνει επανεκκίνηση του συστήματος. |
| **Αυτόματη εκκίνηση** | Πολλές από αυτές τις υπηρεσίες εκκινούν αυτόματα κατά την εκκίνηση του συστήματος (boot). |
| **Διαχείριση μέσω systemd** | Στις περισσότερες μοντέρνες Linux διανομές, οι υπηρεσίες διαχειρίζονται από το systemd, το οποίο είναι το προεπιλεγμένο init σύστημα. Άλλα init συστήματα περιλαμβάνουν το Upstart και το SysVinit. |

Μερικές συνηθισμένες υπηρεσίες στο Linux περιλαμβάνουν:

|  |  |
| --- | --- |
| **Networking** | Π.χ., NetworkManager για την αυτόματη διαχείριση των συνδέσεων δικτύου. |
| **Cron** | Για την εκτέλεση προγραμματισμένων εργασιών σε τακτά χρονικά διαστήματα. |
| **SSH (Secure Shell)** | Για απομακρυσμένη σύνδεση και διαχείριση του συστήματος. |
| **Firewall (π.χ., firewalld)** | Για την προστασία του συστήματος από κακόβουλες συνδέσεις. |

### To init σύστημα

Το init σύστημα είναι το πρώτο πρόγραμμα που εκτελείται κατά την εκκίνηση ενός Linux (ή γενικότερα Unix-based) λειτουργικού συστήματος και αποτελεί τον πρώτο πυρήνα διαχείρισης των διεργασιών (process management) του συστήματος. Το init είναι υπεύθυνο για την εκκίνηση και διαχείριση όλων των υπόλοιπων διεργασιών του συστήματος, και παραμένει σε λειτουργία για να επιτηρεί και να διαχειρίζεται τυχόν νέες διεργασίες καθ’ όλη τη διάρκεια της λειτουργίας του.

Οι βασικές λειτουργίες του init συστήματος είναι:

|  |  |
| --- | --- |
| **Εκκίνηση διεργασιών** | Ξεκινά βασικές υπηρεσίες και δαίμονες όπως η δικτύωση (networking), τα συστήματα αρχείων, η καταγραφή συμβάντων (logging) κ.λπ., ώστε το σύστημα να είναι έτοιμο για χρήση. |
| **Διαχείριση διεργασιών** | Παρακολουθεί την κατάσταση των διεργασιών, φροντίζοντας να επανεκκινεί βασικές υπηρεσίες αν τερματιστούν απροσδόκητα. |
| **Έλεγχος λειτουργιών του συστήματος** | Εκτελεί λειτουργίες που ορίζονται στα αρχεία διαμόρφωσης και διαχειρίζεται το επίπεδο εκτέλεσης (run level) του συστήματος (π.χ., λειτουργία single-user mode, multi-user mode κ.λπ.). |
| **Τερματισμός του συστήματος** | Διαχειρίζεται τον σωστό τερματισμό των διεργασιών και την απόσυρση των υπηρεσιών κατά τον τερματισμό του συστήματος. |

### Είδη init συστημάτων

Υπάρχουν διαφορετικά init συστήματα στο Linux, με το κάθε ένα να προσφέρει διαφορετικά πλεονεκτήματα και λειτουργίες:

1. **SysVinit:**
   * Το κλασικό σύστημα init που χρησιμοποιείται παραδοσιακά στα Unix και Linux συστήματα.
   * Χρησιμοποιεί τα επίπεδα εκτέλεσης (runlevels) για να καθορίσει ποιες διεργασίες θα ξεκινήσουν κατά την εκκίνηση.
   * Είναι πιο απλό, αλλά έχει περιορισμούς, όπως δυσκολία στον χειρισμό περίπλοκων εξαρτήσεων μεταξύ υπηρεσιών.
2. **Upstart:**
   * Δημιουργήθηκε από την Canonical για το Ubuntu ως αντικαταστάτης του SysVinit.
   * Υποστηρίζει δυναμική εκκίνηση διεργασιών βάσει γεγονότων, π.χ., όταν μια συσκευή συνδέεται ή αποσυνδέεται.
   * Ήταν ένα βήμα προς ένα πιο μοντέρνο init σύστημα, αλλά σταδιακά εγκαταλείφθηκε από τις περισσότερες διανομές υπέρ του systemd.
3. **Systemd:**
   * Το πιο διαδεδομένο init σύστημα στις σύγχρονες διανομές Linux (π.χ., Fedora, Ubuntu, Arch Linux).
   * Χρησιμοποιεί μια δομή βασισμένη σε μονάδες (units) και προσφέρει δυνατότητες όπως παράλληλη εκκίνηση διεργασιών, εξαιρετική διαχείριση εξαρτήσεων και τη δυνατότητα παρακολούθησης των υπηρεσιών σε πραγματικό χρόνο.
   * Περιλαμβάνει πολλές επιπλέον λειτουργίες πέραν της εκκίνησης του συστήματος, όπως τη διαχείριση καταγραφών (journald) και την παρακολούθηση διεργασιών.

Το init σύστημα είναι κρίσιμο για τη σταθερότητα και την απόδοση ενός συστήματος Linux. Επιτρέπει την ομαλή εκκίνηση και παρακολούθηση των βασικών λειτουργιών και υπηρεσιών, καθώς και τη διαχείριση των πόρων του συστήματος. Ένα αποτελεσματικό init σύστημα μπορεί να μειώσει τους χρόνους εκκίνησης, να βελτιώσει τη σταθερότητα των υπηρεσιών και να επιτρέψει καλύτερη διαχείριση των εξαρτήσεων, καθιστώντας το Linux πιο προσιτό και λειτουργικό για διαφορετικούς τύπους χρήσεων, από διακομιστές έως επιτραπέζιους υπολογιστές.

## To systemd init σύστημα

Το systemd είναι το init σύστημα που χρησιμοποιείται ευρέως στα περισσότερα σύγχρονα Linux distributions (π.χ., Fedora, Ubuntu, Arch Linux) και έχει αντικαταστήσει τα παλαιότερα συστήματα, όπως το SysVinit και το Upstart. Δημιουργήθηκε με σκοπό να βελτιώσει την εκκίνηση του συστήματος και να παρέχει πλούσια διαχείριση υπηρεσιών και διεργασιών, προσαρμοσμένη στις σύγχρονες ανάγκες και υποδομές.

Παρακάτω εκτίθενται τα κύρια χαρακτηριστικά του systemd:

1. **Παράλληλη εκκίνηση διεργασιών:**
   * Επιτρέπει την εκκίνηση διεργασιών και υπηρεσιών παράλληλα, σε αντίθεση με το παραδοσιακό SysVinit, που ακολουθεί σειριακή εκκίνηση. Αυτό μειώνει τους χρόνους εκκίνησης.
2. **Διαχείριση εξαρτήσεων:**
   * Το systemd διαχειρίζεται εξαρτήσεις μεταξύ υπηρεσιών, εκκινώντας τις με τη σωστή σειρά και ξαναεκκινώντας τυχόν υπηρεσίες που αποτυγχάνουν.
3. **Δομή βασισμένη σε μονάδες (Units):**
   * Στο systemd, κάθε υπηρεσία, mount point, socket ή διεργασία διαχειρίζεται ως μονάδα (unit). Υπάρχουν διαφορετικοί τύποι μονάδων (units), όπως service, socket, mount, target, κ.λπ.
4. **Journald:**
   * Το systemd περιλαμβάνει ένα σύστημα καταγραφής, το journald, που συλλέγει και αποθηκεύει καταγραφές για το σύστημα και τις εφαρμογές.
5. **Διαχείριση λειτουργίας συστήματος:**
   * Επιτρέπει την παρακολούθηση και διαχείριση του συστήματος μέσω εντολών όπως systemctl για τον έλεγχο της κατάστασης των υπηρεσιών και journalctl για την καταγραφή καταχωρίσεων.

### Τα κύρια πλεονεκτήματα του systemd είναι:

* **Γρήγορη εκκίνηση** μέσω παράλληλων διεργασιών.
* **Αξιόπιστη διαχείριση εξαρτήσεων** και αυτόματη επανεκκίνηση υπηρεσιών.
* **Ομοιογενής διαχείριση καταγραφών** με το journald.
* **Καλύτερη διαχείριση πόρων** χάρη στη δυνατότητα ομαδοποίησης διεργασιών.

Οι μονάδες του systemd περιγράφουν κάθε διεργασία ή υπηρεσία που μπορεί να ξεκινήσει, να σταματήσει ή να παρακολουθείται στο σύστημα. Παραδείγματα μονάδων περιλαμβάνουν:

|  |  |
| --- | --- |
| **Service Units** (.service) | Διαχειρίζονται υπηρεσίες, όπως sshd.service. |
| **Target Units** (.target) | Ορίζουν επίπεδα εκτέλεσης και εκτελούν πολλαπλές υπηρεσίες μαζί (π.χ., multi-user.target). |
| **Socket Units** (.socket) | Διαχειρίζονται sockets δικτύου ή διακοπτών (IPC). |
| **Mount Units** (.mount) | Χρησιμοποιούνται για την αυτόματη σύνδεση (mount) των συστημάτων αρχείων. |

Το systemd παρέχει ένα σύγχρονο, επεκτάσιμο και αποδοτικό σύστημα διαχείρισης υπηρεσιών για το Linux, διευκολύνοντας τη διαχείριση του συστήματος και τη σταθερότητα, ακόμα και σε απαιτητικά περιβάλλοντα.

### Service units

Κάθε service unit αρχείο (π.χ., myservice.service) έχει μια βασική δομή που περιγράφει την υπηρεσία, τις ενέργειες εκκίνησης/τερματισμού και τις εξαρτήσεις. Ακολουθεί ένα παράδειγμα αρχείου .service για μία υποθετική υπηρεσία myservice:

# /etc/systemd/system/myservice.service

[Unit]

Description=My Custom Service

After=network.target

[Service]

ExecStart=/usr/bin/myapplication --start

ExecStop=/usr/bin/myapplication --stop

Restart=on-failure

[Install]

WantedBy=multi-user.target

Ανάλυση:

* [Unit]:
  + **Description**: Μια σύντομη περιγραφή της υπηρεσίας.
  + **After**: Καθορίζει τις εξαρτήσεις, π.χ., το network.target δηλώνει ότι η υπηρεσία θα ξεκινήσει αφού η δικτυακή υπηρεσία είναι διαθέσιμη.
* [Service]:
  + **ExecStart:** Ορίζει την εντολή για την εκκίνηση της υπηρεσίας.
  + **ExecStop:** Ορίζει την εντολή για τον τερματισμό της υπηρεσίας.
  + **Restart:** Καθορίζει την πολιτική επανεκκίνησης (εδώ επανεκκινεί σε περίπτωση αποτυχίας).
* [Install]:
  + **WantedBy:** Ορίζει τον στόχο εκκίνησης στον οποίο θα συνδεθεί η υπηρεσία (π.χ., multi-user.target).

### Target units

Τα target unitαρχεία στο systemd είναι μονάδες που αντιπροσωπεύουν συλλογές άλλων units και ουσιαστικά ομαδοποιούν υπηρεσίες και λειτουργίες του συστήματος, καθορίζοντας την κατάσταση λειτουργίας του. Τα target units χρησιμοποιούνται κυρίως για να επιτυγχάνεται μια συγκεκριμένη κατάσταση λειτουργίας του συστήματος, όπως το multi-user mode ή το graphical mode, αντίστοιχα με τα παραδοσιακά runlevels που υπήρχαν στο SysVinit.

Tα κύρια χαρακτηριστικά των target unit αρχείων είναι τα κάτωθι:

1. **Ομαδοποίηση διεργασιών**: Κάθε target unit συνδέει και ενεργοποιεί άλλες μονάδες, όπως service units, mount units, και άλλα target units. Έτσι, επιτυγχάνεται εύκολη διαχείριση πολλών υπηρεσιών μέσω μιας μονάδας.
2. **Διαχείριση εξαρτήσεων:** Τα target units προσφέρουν μια δομή εξαρτήσεων, δηλαδή οι υπηρεσίες που χρειάζονται για να φτάσει το σύστημα σε μια συγκεκριμένη κατάσταση εκκινούνται αυτόματα. Έτσι, το σύστημα διασφαλίζει ότι όλες οι σχετικές υπηρεσίες θα τρέξουν με την σωστή σειρά.
3. **Διαδοχικές καταστάσεις εκτέλεσης**: Το systemd μπορεί να εκκινήσει διαφορετικά target units σε διαφορετικά στάδια εκκίνησης, επιτρέποντας τη σταδιακή ενεργοποίηση των υπηρεσιών και των πόρων του συστήματος.

Τα κύρια target units εκτίθενται παρακάτω:

1. **basic.target:** Ορίζει το ελάχιστο λειτουργικό περιβάλλον του συστήματος και εξαρτάται από βασικές υπηρεσίες. Χρησιμοποιείται ως σημείο εκκίνησης για άλλες λειτουργίες.
2. **multi-user.target:** Ενεργοποιεί το multi-user mode και όλες τις απαραίτητες υπηρεσίες για χρήση σε περιβάλλον πολλαπλών χρηστών (χωρίς γραφικό περιβάλλον). Είναι το αντίστοιχο του runlevel 3 στα παραδοσιακά Unix συστήματα.
3. **graphical.target:** Ενεργοποιεί το γραφικό περιβάλλον (GUI) και είναι το προεπιλεγμένο επίπεδο για συστήματα με επιφάνεια εργασίας. Είναι το αντίστοιχο του runlevel 5 στο παλιό σύστημα.
4. **reboot.target:** Ορίζει το στόχο επανεκκίνησης του συστήματος. Όταν καλείται, προκαλεί επανεκκίνηση του συστήματος.
5. **shutdown.target:** Σχετίζεται με τον τερματισμό του συστήματος, εξασφαλίζοντας ότι όλες οι υπηρεσίες θα σταματήσουν πριν το σύστημα τερματιστεί.
6. **rescue.target**: Ενεργοποιεί το rescue mode, το οποίο παρέχει βασική λειτουργία για επισκευή και αποκατάσταση. Είναι το αντίστοιχο του single-user mode και χρησιμοποιείται για αποκατάσταση προβλημάτων συστήματος.
7. **emergency.target**: Αποτελεί τη βάση για το emergency mode, το οποίο παρέχει έναν ελάχιστο περιβάλλον για την αντιμετώπιση σοβαρών προβλημάτων στο σύστημα, χωρίς καμία εξάρτηση από άλλες υπηρεσίες.

Το παρακάτω είναι ένα παράδειγμα ενός custom target unit αρχείου, custom.target, το οποίο ορίζει ένα περιβάλλον που ενεργοποιεί συγκεκριμένες υπηρεσίες:

# /etc/systemd/system/custom.target

[Unit]

Description=Custom Target for Specific Services

Requires=network.target

Wants=sshd.service

After=network.target

[Install]

WantedBy=multi-user.target

Ανάλυση:

* **Description**: Μια σύντομη περιγραφή του target.
* **Requires:** Ορίζει ότι το network.target είναι απαραίτητο για την εκκίνηση του custom.target.
* **Wants:** Το sshd.service δεν είναι απολύτως απαραίτητο, αλλά θα εκκινήσει αν είναι διαθέσιμο.
* **After**: Εξασφαλίζει ότι το custom.target θα ξεκινήσει μετά το network.target.
* **WantedBy**: Καθορίζει ότι το custom.target θα μπορεί να ενεργοποιείται με το multi-user.target.

### Socket units

Τα socket units στο systemd είναι μονάδες που διαχειρίζονται sockets και βοηθούν στον χειρισμό υπηρεσιών δικτύου ή IPC (Inter-Process Communication). Ενεργοποιούν υπηρεσίες μόνο όταν είναι αναγκαίο, δηλαδή όταν έρθει κάποια αίτηση μέσω του socket, εξοικονομώντας έτσι πόρους και βελτιώνοντας την απόδοση.

Η χρήση των socket units είναι ιδιαίτερα χρήσιμη σε servers και περιβάλλοντα με περιορισμένους πόρους, καθώς οι υπηρεσίες δεν τρέχουν συνεχώς, αλλά μόνο όταν υπάρχει αίτημα σύνδεσης. Ουσιαστικά, το systemd ακούει στο socket για εισερχόμενες συνδέσεις και, όταν λάβει κάποιο αίτημα, εκκινεί την κατάλληλη υπηρεσία για να το διαχειριστεί.

Τα socket units συνδέονται άμεσα με service units. Όταν μια υπηρεσία χρειάζεται να διαχειριστεί αιτήματα, το systemd ανοίγει ένα socket και περιμένει. Αν έρθει μια αίτηση στο socket, το systemd ενεργοποιεί το αντίστοιχο service unit, παρέχοντας στο socket τον έλεγχο της υπηρεσίας.

Η συσχέτιση socket-service επιτρέπει την «ενεργοποίηση κατ’ απαίτηση» (on-demand activation), όπου η υπηρεσία ξεκινάει μόνο όταν υπάρχει αίτηση.

Τα socket units ακολουθούν συγκεκριμένη δομή και περιλαμβάνουν παραμέτρους που ορίζουν τον τύπο του socket, τη διεύθυνση, το πρωτόκολλο και άλλες ιδιότητες. Ένα τυπικό .socket αρχείο έχει τις εξής ενότητες:

1. **[Unit]:** Περιέχει πληροφορίες για το unit και τις εξαρτήσεις του.
2. **[Socket]:** Ρυθμίσεις για το socket, όπως το είδος και η διεύθυνση.
3. **[Install]:** Ρυθμίσεις για την εγκατάσταση και ενεργοποίηση της μονάδας.

Ακολουθεί ένα παράδειγμα αρχείου .socket για μια υποθετική υπηρεσία myservice που ακούει για TCP συνδέσεις στην πόρτα 8080.

# /etc/systemd/system/myservice.socket

[Unit]

Description=Socket for My Custom Service

[Socket]

ListenStream=8080

Accept=no

[Install]

WantedBy=sockets.target

Και το αντίστοιχο αρχείο service για το myservice:

# /etc/systemd/system/myservice.service

[Unit]

Description=My Custom Service

After=network.target

[Service]

ExecStart=/usr/bin/myservice

Ανάλυση:

* ListenStream=8080: Ορίζει ότι το socket ακούει στην TCP πόρτα 8080.
* Accept=no: Καθορίζει ότι το socket δεν δημιουργεί νέο instance της υπηρεσίας για κάθε σύνδεση, αλλά κρατά ένα διαμπορασμένο instance.
* WantedBy=sockets.target: Καθορίζει ότι το socket θα ενεργοποιείται με το sockets.target.

Τα κύρια οφέλη των socket units είναι:

|  |  |
| --- | --- |
| **Εξοικονόμηση Πόρων**: | Επιτρέπουν την εκκίνηση υπηρεσιών μόνο όταν υπάρχει ανάγκη, αντί να τρέχουν συνεχώς στο παρασκήνιο. |
| **Ασφάλεια και Εξοικονόμηση Χρόνου**: | Τα sockets διαχειρίζονται με ενοποιημένο τρόπο από το systemd, μειώνοντας την πολυπλοκότητα στη διαχείριση και τον έλεγχο πρόσβασης. |
| **Καλύτερη Ενσωμάτωση με το Σύστημα**: | Διευκολύνει τη διαχείριση υπηρεσιών και εξαρτήσεων, με αποτέλεσμα πιο αξιόπιστη και ασφαλή λειτουργία. |

Τα socket units είναι έτσι βασικός μηχανισμός για την αποδοτική διαχείριση δικτυακών υπηρεσιών και επικοινωνίας διεργασιών στο systemd.

### Mount units

Τα mount units στο systemd είναι μονάδες που διαχειρίζονται τα *mount points* στο σύστημα αρχείων. Αυτές οι μονάδες επιτρέπουν στο systemd να διαχειρίζεται την προσάρτηση (mounting) των συσκευών και των file systems, όπως οι δίσκοι και τα εξωτερικά μέσα αποθήκευσης, σύμφωνα με μια σειρά κανόνων και εξαρτήσεων.

Ένα mount unit είναι στην ουσία το ανάλογο μιας γραμμής στο αρχείο /etc/fstab, με το πρόσθετο πλεονέκτημα ότι προσφέρει μεγαλύτερη ευελιξία στη διαχείριση και στην αυτοματοποίηση της προσάρτησης συσκευών και συστημάτων αρχείων.

Κάθε mount unit έχει τη δική του ρύθμιση, η οποία αποθηκεύεται σε αρχείο με κατάληξη .mount. Το όνομα του αρχείου καθορίζεται από το mount point, με αντικατάσταση των χαρακτήρων / με -. Για παράδειγμα, το mount point /mnt/data αντιστοιχεί στο αρχείο μονάδας mnt-data.mount.

Ένα αρχείο .mount έχει τις εξής βασικές ενότητες:

1. **[Unit]:** Περιλαμβάνει πληροφορίες για το unit και τις εξαρτήσεις του.
2. **[Mount]:** Ρυθμίσεις για το mount, όπως η τοποθεσία και οι επιλογές προσάρτησης.
3. **[Install]:** Περιέχει ρυθμίσεις για την ενεργοποίηση της μονάδας κατά την εκκίνηση του συστήματος.

Ακολουθεί ένα παράδειγμα για να προσδεθεί ένας εξωτερικός δίσκος στο /mnt/data:

# /etc/systemd/system/mnt-data.mount

[Unit]

Description=Mount External Drive to /mnt/data

After=network.target

[Mount]

What=/dev/sdb1

Where=/mnt/data

Type=ext4

Options=defaults

[Install]

WantedBy=multi-user.target

Ανάλυση:

* What=/dev/sdb1: Ορίζει τη συσκευή που θα προσαρτηθεί (π.χ., /dev/sdb1).
* Where=/mnt/data: Το σημείο προσάρτησης.
* Type=ext4: Το σύστημα αρχείων (π.χ., ext4).
* Options=defaults: Ορίζει επιλογές προσάρτησης. Μπορεί να περιέχει επιλογές όπως rw, noatime, κ.λπ.
* WantedBy=multi-user.target: Καθορίζει ότι η μονάδα θα είναι ενεργή στο multi-user mode.

Τα mount units παρέχουν έναν εξαιρετικά αποδοτικό τρόπο διαχείρισης της προσάρτησης συσκευών και συστήματα αρχείων στο systemd, επιτρέποντας βελτιωμένη αυτοματοποίηση και μεγαλύτερη ευελιξία σε σύγκριση με τη μονολιθική πρακτική του /etc/fstab.

## Δαίμονες (Daemons)

Οι δαίμονες (daemons) είναι ειδικοί τύποι υπηρεσιών που εκτελούνται στο παρασκήνιο και συνήθως δεν απαιτούν καμία άμεση αλληλεπίδραση με τον χρήστη. Πολλοί δαίμονες ξεκινούν κατά την εκκίνηση του συστήματος και συνεχίζουν να λειτουργούν για να υποστηρίζουν συγκεκριμένες εργασίες, όπως η παρακολούθηση της κατάστασης του συστήματος, η καταγραφή δεδομένων και η εξυπηρέτηση αιτήσεων δικτύου.

#### Τα κύρια χαρακτηριστικά τους είναι:

|  |  |
| --- | --- |
| **Μακροχρόνια λειτουργία** | Παραμένουν σε λειτουργία χωρίς την ανάγκη για επανεκκίνηση ή παρέμβαση από τον χρήστη. |
| **Ανταποκρίνονται σε αιτήσεις ή εκδηλώσεις** | Π.χ., ο δαίμονας httpd (Apache HTTP Server) ανταποκρίνεται σε αιτήσεις ιστοσελίδων. |
| **Αναγνωρίζονται με το γράμμα “d”** | Οι περισσότεροι δαίμονες τελειώνουν με το γράμμα “d” στο όνομά τους (π.χ., httpd, sshd, crond). |

Μερικοί βασικοί δαίμονες οι οποίοι χρησιμοποιούνται ευρέως είναι:

|  |  |
| --- | --- |
| **sshd** | Διαχειρίζεται τις απομακρυσμένες συνδέσεις μέσω SSH. |
| **httpd ή apache2** | Διαχειρίζεται τις υπηρεσίες του Web Server Apache. |
| **crond** | Εκτελεί προγραμματισμένες εργασίες. |
| **systemd-logind** | Διαχειρίζεται τις συνδέσεις χρηστών και την παρακολούθηση της κατάστασης λειτουργίας. |
| **cupsd** | Διαχειρίζεται τις εργασίες εκτύπωσης και τον εκτυπωτή. |

Η διαχείριση των υπηρεσιών και των δαιμόνων στο Linux γίνεται συνήθως μέσω του systemd, του κυριότερου init συστήματος στις σύγχρονες διανομές, το οποίο επιτρέπει στους διαχειριστές να ξεκινούν, να σταματούν, να επανεκκινούν, να ενεργοποιούν και να απενεργοποιούν τις υπηρεσίες και τους δαίμονες.

### Η εντολή systemctl

H εντολή systemctl είναι το εργαλείο διαχείρισης υπηρεσιών του systemd στο Linux και επιτρέπει τον έλεγχο και τη διαχείριση των δαιμόνων που εκτελούνται ως υπηρεσίες στο παρασκήνιο. Η χρήση της systemctl περιλαμβάνει ένα σύνολο εντολών για την εκκίνηση, διακοπή, επανεκκίνηση και έλεγχο της κατάστασης των *daemons*, καθώς και για την εκτέλεση διαχειριστικών ενεργειών.

Συνοπτικά οι χρήσεις της systemctl είναι:

|  |  |
| --- | --- |
| **Εκκίνηση:** | systemctl start <daemon>.service |
| **Διακοπή:** | systemctl stop < daemon >.service |
| **Επανεκκίνηση:** | systemctl restart < daemon >.service |
| **Έλεγχος κατάστασης:** | systemctl status <daemon>.service |
| **Ενεργοποίηση κατά την εκκίνηση:** | systemctl enable <daemon>.service |
| **Απενεργοποίηση κατά την εκκίνηση:** | systemctl disable <daemon>.service |

Αναλυτικότερα με την systemctl μπορούμε να επιτελέσουμε τα κάτωθι:

* **Έναρξη (start).** Η εντολή systemctl start <daemon> χρησιμοποιείται για την εκκίνηση ενός δαίμονα. Συγεκριμένα για να εκκινήσουμε τον Apache httpd δαίμονα, μπορούμε να εκδώσουμε την κάτωθι εντολή:

systemctl start httpd.service

*Σημειωση: Σε μερικές διανομές ο* httpd *δαίμονας μπορεί να λέγεται και* apache2.

* **Διακοπή (stop)**. Η εντολή systemctl stop <daemon> διακόπτει έναν δαίμονα που τρέχει. Αυτό είναι χρήσιμο για τη διακοπή της λειτουργίας του δαίμονα χωρίς να τον διαγράψουμε ή να τον απενεργοποιήσουμε. Συγεκριμένα για να διακόψουμε τον Apache httpd δαίμονα, μπορούμε να εκδώσουμε την κάτωθι εντολή:

systemctl stop httpd.service

* **Επανεκκίνηση (restart).** Η εντολή systemctl restart <daemon> χρησιμοποιείται για να σταματήσει και να ξεκινήσει ξανά έναν δαίμονα. Αυτό είναι χρήσιμο για την εφαρμογή αλλαγών ρυθμίσεων ή την ανανέωση της υπηρεσίας. Συγεκριμένα για να επανεκκινήσουμε τον Apache httpd δαίμονα, μπορούμε να εκδώσουμε την κάτωθι εντολή:

systemctl restart httpd.service

* **Έλεγχος κατάστασης (status).** Η εντολή systemctl status <daemon> παρέχει πληροφορίες σχετικά με την τρέχουσα κατάσταση ενός δαίμονα. Εμφανίζει αν ο δαίμονας είναι ενεργός, σταματημένος ή αν υπάρχουν σφάλματα.

Ενδεικτικά, η εντολή systemctl status postfix δίνει το παρακάτω αποτέλεσμα για την κατάσταση του Postfix Μail Τransport Αgent:

postfix.service - Postfix Mail Transport Agent

Loaded: loaded (/lib/systemd/system/postfix.service; enabled)

Active: active (exited) since Wed 2024-07-03 21:55:18 EEST;

3 months 28 days ago

Main PID: 2330 (code=exited, status=0/SUCCESS)

Tasks: 0 (limit: 4522)

CGroup: /system.slice/postfix.service

* **Ενεργοποίηση/Απενεργοποίηση κατά την εκκίνηση του συστήματος (enable και disable).** Η εντολή systemctl enable <daemon> διασφαλίζει ότι ο δαίμονας θα ξεκινήσει αυτόματα με την εκκίνηση του συστήματος.

Η εντολή systemctl disable <daemon> διασφαλίζει ότι ο δαίμονας δεν θα ξεκινήσει αυτόματα με την εκκίνηση του συστήματος.

* **Αναγκαστική επανεκκίνηση (restart --force).** Σε περιπτώσεις όπου μια υπηρεσία δεν ανταποκρίνεται, η αναγκαστική επανεκκίνηση μπορεί να βοηθήσει. Η εντολή systemctl restart --force <daemon> εξαναγκάζει την επανεκκίνηση, αλλά πρέπει να χρησιμοποιείται με προσοχή.
* **Mask και Unmask.** Η εντολή systemctl mask <daemon> αποτρέπει την εκκίνηση του δαίμονα, ακόμη και αν κάποιος προσπαθήσει να τον εκκινήσει χειροκίνητα. Η ενέργεια αυτή δημιουργεί έναν σύνδεσμο που κάνει τον εν λόγω δαίμονα μη προσβάσιμο. Αν χρειαστεί να επαναφέρουμε την πρόσβαση, χρησιμοποιούμε την εντολή systemctl unmask <daemon>. Αυτό αφαιρεί τον περιορισμό, και ο επηρεαζόμενος δαίμονας μπορεί να εκκινήσει κανονικά.
* **Έλεγχος καταγραφών με το journalctl**. Μέσω της εντολής journalctl, μπορούμε να δούμε τα logs του systemd για συγκεκριμένους δαίμονες:

journalctl -u httpd.service

* **Εντολές για πολλαπλές υπηρεσίες.** Η systemctl επιτρέπει επίσης τη διαχείριση πολλών υπηρεσιών ταυτόχρονα, χρησιμοποιώντας *wildcards:*

systemctl restart '\*.service'

* **Επαναφόρτωση της διαμόρφωσης**. Η εντολή systemctl reload <daemon> επαναφορτώνει τη διαμόρφωση ενός δαίμονα χωρίς να τον σταματήσει και να τον ξεκινήσει πάλι, εφόσον ο εν λόγω δαίμονας υποστηρίζει τη δυνατότητα αυτή. Είναι χρήσιμη όταν θέλουμε να εφαρμόσουμε νέες ρυθμίσεις χωρίς διακοπή της λειτουργίας του δαίμονα.

systemctl reload httpd.service

Η systemctl καθιστά εύκολη την κεντρική διαχείριση δαιμόνων στο Linux, προσφέροντας ολοκληρωμένες δυνατότητες παρακολούθησης και ελέγχου, που εξασφαλίζουν τη σωστή λειτουργία και ασφάλεια των υπηρεσιών του συστήματος.

### Προβλήματα κατά την εκκίνηση δαιμόνων

Όταν ξεκινάμε δαίμονες με την systemctl, μπορεί να προκύψουν διάφορα θέματα. Ακολουθούν μερικά από τα πιο κοινά προβλήματα και οι τρόποι επίλυσής τους:

* **Σφάλματα δικαιωμάτων.** Εάν ο δαίμονας χρειάζεται αυξημένα δικαιώματα και προσπαθούμε να τον εκκινήσουμε ως μη-διαχειριστής, μπορεί να λάβουμε σφάλμα πρόσβασης.

**Λύση**: Εκκινείστε τη διαδικασία με sudo, όπως sudo systemctl start daemon, ή βεβαιωθείτε ότι ο χρήστης έχει τα απαραίτητα δικαιώματα.

* **Ελλείποντα αρχεία διαμόρφωσης ή εξαρτήσεις.** Αν ένα αρχείο διαμόρφωσης του δαίμονα λείπει ή δεν είναι σωστά ρυθμισμένο, ο δαίμονας μπορεί να μην ξεκινήσει καθόλου.

**Λύση**: Ελέγχετε τα αρχεία διαμόρφωσης που απαιτούνται για τον εκ λόγω δλαιμονα

* **Αναφορά σε λάθος εκτελέσιμα αρχεία.** Σε ορισμένες περιπτώσεις, το αρχείο .service μπορεί να αναφέρεται σε λάθος διαδρομή εκτελέσιμου αρχείου.

**Λύση**: Επανελέγχετε το .service αρχείο και επιβεβαιώνουμε ότι το ExecStart δείχνει στο σωστό εκτελέσιμο αρχείο.

* **Ο δαίμονας είναι ήδη σε εκτέλεση.** Αν ο δαίμονας εκτελείται ήδη και προσπαθήσουμε να τον ξεκινήσουμε, θα λάβουμε μήνυμα ότι δεν μπορεί να ξεκινήσει διότι ήδη τρέχει.

**Λύση**: Χρησιμοποίηστε την εντολή systemctl restart daemon αντί για start για να επανεκκινήσουμε τον δαίμονα.

* **Πρόβλημα με τις εξαρτήσεις του δαίμονα.** Ορισμένοι δαίμονες χρειάζονται άλλες υπηρεσίες να είναι ενεργές πρώτα για να λειτουργήσουν σωστά.

**Λύση**: Έλεγξε τα αρχεία διαμόρφωσης .service και βεβαιώσου ότι οι εξαρτήσεις ορίζονται σωστά με τις ρυθμίσεις After= και Requires=.

* **Μη έγκυρη διαμόρφωση του αρχείου systemd.** Αν το αρχείο .service περιέχει συντακτικά λάθη ή ακατάλληλες παραμέτρους, μπορεί να προκαλέσει σφάλματα κατά την εκκίνηση.

**Λύση**: Τρέξτε systemd-analyze verify /path/to/service/file για να ελέγξεις για συντακτικά λάθη.

* **Καθυστέρηση εκκίνησης λόγω χρονικού ορίου (Timeout)**. Ορισμένες φορές, αν ο δαίμονας δεν απαντά γρήγορα, η systemctl μπορεί να αποτύχει με σφάλμα χρονικού ορίου.

**Λύση**: Προσαρμόστε την παράμετρο TimeoutStartSec στο αρχείο .service, αν χρειάζεται περισσότερος χρόνος για εκκίνηση.

* **Σφάλμα επικοινωνίας με το journald**. Αν ο δαίμονας προσπαθεί να καταγράψει σε ένα κλειστό κανάλι στο journald, μπορεί να προκύψει αποτυχία εκκίνησης.

**Λύση**: Επανεκκίνηστε το systemd-journald χρησιμοποιώντας systemctl restart systemd-journald. Αν παρ’ όλα αυτά αντιμετωπίζετε προβλήματα, η εντολή journalctl -xe ή systemctl status service\_name δίνει πιο λεπτομερείς πληροφορίες για τη διάγνωση και επίλυση των προβλημάτων.

### Δικτυακές υπηρεσίες

Το systemd παρέχει ένα ισχυρό σύνολο εργαλείων για τη διαχείριση και την παρακολούθηση δικτυακών υπηρεσιών και δαιμόνων σε ένα σύστημα Linux, επιτρέποντας στους διαχειριστές να ελέγχουν και να εποπτεύουν κρίσιμες υπηρεσίες όπως οι εξής:

1. **SSH (Secure Shell)** - Ο δαίμονας sshd.service προσφέρει πρόσβαση στο σύστημα μέσω του πρωτοκόλλου SSH. Με το systemctl, μπορούμε να τον παρακολουθούμε και να τον επανεκκινούμε ή να τον σταματούμε για λόγους ασφαλείας ή συντήρησης:

systemctl status sshd.service

systemctl restart sshd.service

1. **Apache HTTP Server** - Ο web server apache2.service (ή httpd.service σε Red Hat-based διανομές) εξυπηρετεί αιτήσεις HTTP, επιτρέποντας τη φιλοξενία ιστοσελίδων. Είναι διαχειρίσιμος με το systemd και μπορεί να παρακολουθείται για στατιστικά χρήσης και σφάλματα.

systemctl status apache2.service

systemctl restart apache2.service

1. **MySQL/MariaDB** - Αυτές οι βάσεις δεδομένων λειτουργούν μέσω των δαιμόνων mysql.service ή mariadb.service. Το systemd επιτρέπει στους διαχειριστές να ελέγχουν την κατάσταση και να επανεκκινούν τον δαίμονα όταν χρειάζεται:

systemctl status mysql.service

systemctl restart mysql.service

1. **Nginx** - Ο δαίμονας nginx.service για τον web server Nginx, που διαχειρίζεται τις αιτήσεις HTTP και reverse proxy, ελέγχεται και παρακολουθείται μέσω του systemd.

systemctl status nginx.service

systemctl restart nginx.service

1. **Docker** - Ο δαίμονας docker.service διαχειρίζεται απομονωτές (containers) και μπορεί να παρακολουθείται ή να επανεκκινείται μέσω του systemd για τη διαχείριση των υπηρεσιών απομονωτών.

systemctl status docker.service

systemctl restart docker.service

Με αυτά τα εργαλεία, το systemd γίνεται ένα ισχυρό εργαλείο για τη συνεχή παρακολούθηση και διαχείριση δικτυακών υπηρεσιών, διασφαλίζοντας την αξιοπιστία και τη σταθερότητα των συστημάτων.

## Διαφορές μεταξύ υπηρεσιών και δαιμόνων

Στο Linux, οι όροι **υπηρεσία** και **δαίμονας** (**daemon)** σχετίζονται με διεργασίες που εκτελούνται στο παρασκήνιο, αλλά έχουν κάποιες διαφορές στην έννοια και στη χρήση τους, συγκεκριμένα:.

1. **Daemon**: Ένας δαίμονας είναι μια ανεξάρτητη διαδικασία που τρέχει στο παρασκήνιο και εξυπηρετεί συγκεκριμένες λειτουργίες, όπως να δέχεται αιτήματα από χρήστες ή άλλες εφαρμογές (π.χ., sshd για SSH, httpd για HTTP). Οι δαίμονες τρέχουν χωρίς άμεση αλληλεπίδραση από τον χρήστη και συνήθως ξεκινούν κατά την εκκίνηση του συστήματος. Ένας δαίμονας εκκινεί ανεξάρτητα και μένει στο παρασκήνιο μέχρι να λάβει κάποιο αίτημα ή να τερματιστεί χειροκίνητα. Πολλές φορές, ένας δαίμονας συνεχίζει να λειτουργεί ακόμη κι αν το σύστημα δεν υποστηρίζει την αυτόματη διαχείριση υπηρεσιών. Επίσης, οι δαίμονες δεν έχουν εξαρτήσεις από άλλες υπηρεσίες ή από το init σύστημα για να ξεκινήσουν. Τουναντίον, μπορούν να εκτελούνται ανεξάρτητα και αδιάλλειπτα.
2. **Υπηρεσία**: Μια υπηρεσία είναι μια λειτουργικότητα ή εφαρμογή που διαχειρίζεται από το init σύστημα, όπως το systemd. Μπορεί να είναι ένας daemon αλλά και κάτι περισσότερο: Mια υπηρεσία μπορεί να είναι μια οποιαδήποτε διαδικασία που το σύστημα παρακολουθεί και διαχειρίζεται ως μέρος της λειτουργίας του. Το systemd παρέχει τον έλεγχο των υπηρεσιών με διαχείριση κατάστασης (π.χ., start, stop, enable, disable). Υπόκειται σε έλεγχο από το systemd ή άλλο σύστημα init. Η υπηρεσία μπορεί να επανεκκινηθεί αυτόματα, να τερματιστεί ελεγχόμενα, και να παρακολουθείται συνεχώς για αποτυχίες.

Συνοψίζοντας, η κύρια διαφορά μεταξύ των δύο έγκειται στην ενσωμάτωση των υπηρεσιών στο σύστημα διαχείρισης και ελέγχου του systemd. Αντίθετα, οι δαίμονες επικεντρώνονται σε μία συγκεκριμένη λειτουργία ή υπηρεσία χωρίς απαραίτητα να διαχειρίζονται μέσω του συστήματος init.

## Άσκηση: Παρακολούθηση και έλεγχος δικτυακών υπηρεσιών και υπηρεσιών συστήματος

Σας έχει ανατεθεί η ευθύνη για τη διαχείριση ενός διακομιστή που παρέχει δικτυακές υπηρεσίες όπως web server (Apache/Nginx) και απομακρυσμένης πρόσβασης ssh. Πρέπει να:

1. Ελέγξετε την κατάσταση των υπηρεσιών.
2. Διασφαλίσετε ότι λειτουργούν σωστά και επανεκκινούν αυτόματα αν παρουσιαστεί πρόβλημα.
3. Παρακολουθήσετε τις συνδέσεις δικτύου και τη δραστηριότητα των υπηρεσιών σε πραγματικό χρόνο.

**Σκοπός:** Η άσκηση έχει ως στόχο να εξοικειωθείτε με τη διαδικασία παρακολούθησης και διαχείρισης δικτυακών υπηρεσιών και υπηρεσιών συστήματος μέσω των εργαλείων που παρέχονται από το Linux.

**Προσδοκώμενα αποτελέσματα**: Με την ολοκλήρωση της άσκησης, θα έχετε αποκτήσει πρακτικές γνώσεις για τη διαχείριση δικτυακών υπηρεσιών και υπηρεσιών συστήματος, διασφαλίζοντας τη σταθερότητα και την αξιοπιστία τους.

##### Έλεγχος κατάστασης των Apapche και SSH:

##### Ενεργοποίηση αυτόματης εκκίνησης των Apapche και SSH:

##### Παρακολούθηση Καταγραφών (logs) των Apapche και SSH:

##### Έλεγχος των TCP και UDP συνέσεων:

##### Με την εντολή htop:

##### Εντοπίστε ποια διεργασία καταναλώνει τους περισσότερους πόρους CPU ή μνήμης στο σύστημά σας.

1. Πώς μπορούμε να τερματίσουμε μια διεργασία;

### Ενδεικτική επίλυση

##### Έλεγχος κατάστασης των Apache και SSH:

systemctl status apache2

systemctl status ssh

##### Ενεργοποίηση αυτόματης εκκίνησης των Apache και SSH:

systemctl enable apache2

systemctl enable ssh

##### Παρακολούθηση καταγραφών (logs) των Apache και SSH:

journalctl -u apache2

journalctl -u ssh

##### Έλεγχος των TCP και UDP συνέσεων:

netstat –tulnp

με τους εξής διακόπτες:

* + 1. **t:** TCP συνδέσεις
    2. **u:** UDP συνδέσεις
    3. **l:** Listening sockets
    4. **n:** Εμφάνιση IP διευθύνσεων και θυρών
    5. **p:** Εμφάνιση των διεργασιών

##### Με την εντολή htop

##### Εντοπίστε ποια διεργασία καταναλώνει τους περισσότερους πόρους CPU ή μνήμης στο σύστημά σας.

*Ανοίγοντας το htop και ταξινομώντας τις διεργασίες βάσει της χρήσης CPU ή μνήμης (πλήκτρο* ***F6****), μπορούμε να δούμε ποια διεργασία καταναλώνει τους περισσότερους πόρους. Οι διεργασίες που βρίσκονται στην κορυφή της λίστας είναι αυτές με τη μεγαλύτερη κατανάλωση.*

1. Πώς μπορούμε να τερματίσουμε μια διεργασία;

*Επιλέγουμε την διεργασία και εν συνεχεία πατάμε το πλήκτρο* ***F9*** *(Kill) για να τερματίσουμε την επιλεγμένη διεργασία.*

## Ερωτήσεις αυτοαξιολόγησης

1. Ποια εντολή χρησιμοποιείται για να εκκινήσετε μια υπηρεσία στο systemd;

**Α**.systemctl enable

**B.** systemctl start

**Γ.** systemctl load

Δ**.** systemctl launch

1. Ποια εντολή systemd αποτρέπει μια υπηρεσία από το να εκκινήσει χειροκίνητα ή αυτόματα;

**A.** systemctl stop

**B.** systemctl disable

**Γ.** systemctl mask

**Δ.** systemctl lock

1. Ποια εντολή εμφανίζει πληροφορίες κατάστασης για μία υπηρεσία στο systemd;

**A.** systemctl info

**B.** systemctl log

**Γ.** systemctl status

**Δ.** systemctl show

1. Ποια από τις παρακάτω επιλογές είναι σωστή για την επανεκκίνηση ενός δάιμονα μέσω του systemd;

**A.** systemctl restart <daemon>.service

**B.** systemctl refresh <daemon>.service

**Γ.** systemctl reboot <daemon>.service

**Δ.** systemctl load <daemon>.service

1. Ποιο εργαλείο καταγραφής χρησιμοποιείται από το systemd για τη συλλογή logs;

**A.** logrotate

**B.** syslog

**Γ.** journald

**Δ.** dmesg

1. Ποια από τις παρακάτω εντολές χρησιμοποιείται για να απενεργοποιήσει μια υπηρεσία κατά την εκκίνηση του συστήματος;

**A.** systemctl stop

**B. systemctl disable**

**Γ.** systemctl hide

**Δ.** systemctl remove

1. Ποια εντολή εμφανίζει τα καταγεγραμμένα συμβάντα για μία υπηρεσία στο systemd;

**A.** journalctl -s <daemon>.service

**B.** systemctl logs <daemon>.service

**Γ.** journalctl -u <daemon>.service

**Δ.** systemctl log -s <daemon>.service

1. Ποια είναι η χρήση του systemctl enable <daemon>.service;

**A.** Εκκινεί την υπηρεσία.

**B.** Προσθέτει την υπηρεσία στη λίστα αυτόματης εκκίνησης κατά την εκκίνηση του συστήματος.

**Γ.** Τερματίζει την υπηρεσία.

**Δ.** Ελέγχει την κατάσταση της υπηρεσίας.

1. Τι κάνει η εντολή systemctl reload <service>.service;

**A.** Τερματίζει και εκκινεί ξανά την υπηρεσία.

**B.** Επαναφορτώνει τη διαμόρφωση χωρίς να τερματίσει την υπηρεσία.

**Γ.** Απενεργοποιεί την υπηρεσία από την εκκίνηση.

**Δ.** Προσθέτει την υπηρεσία στη λίστα εκκίνησης.

1. Ποια είναι η εντολή για να σταματήσετε μια ενεργή υπηρεσία;  
   A. systemctl shutdown <daemon>.service  
   B. systemctl disable <daemon>.service   
   Γ**.** systemctl stop <daemon>.service  
   Δ. systemctl kill <daemon>.service

### Απαντήσεις

* + - 1. Β
      2. Γ
      3. Γ
      4. Α
      5. Γ
      6. Β
      7. Γ
      8. Β
      9. Β
      10. Γ

# Διαχείριση και ασφάλιση του SSH – Ανάλυση και αποθήκευση των συμβάντων καταγραφής

## Σύνδεση σε απομακρυσμένο υπολογιστή με SSH

Το **SSH (Secure Shell)** είναι ένα πρωτόκολλο ασφαλούς επικοινωνίας που επιτρέπει στους χρήστες να συνδέονται και να διαχειρίζονται απομακρυσμένα συστήματα μέσω δικτύου. Είναι ιδιαίτερα χρήσιμο σε περιβάλλοντα διαχείρισης server, καθώς παρέχει έναν ασφαλή τρόπο για να εκτελούνται εντολές, να μεταφέρονται αρχεία ή να γίνονται αλλαγές στο σύστημα χωρίς την ανάγκη φυσικής πρόσβασης στον υπολογιστή.

Μια από τις κύριες δυνατότητες του SSH είναι η ασφάλεια που παρέχει μέσω κρυπτογράφησης. Όταν συνδέεσαι σε έναν απομακρυσμένο υπολογιστή, όλα τα δεδομένα που ανταλλάσσονται μεταξύ του τοπικού και του απομακρυσμένου συστήματος είναι κρυπτογραφημένα. Αυτό εξασφαλίζει ότι κανείς δεν μπορεί να υποκλέψει τις πληροφορίες (όπως κωδικούς πρόσβασης ή εντολές) κατά τη διάρκεια της μετάδοσης.

Επιπλέον, το SSH προσφέρει άλλες δυνατότητες ασφαλείας, όπως η επαλήθευση της ταυτότητας του απομακρυσμένου υπολογιστή, προκειμένου να αποφευχθεί η υποκλοπή δεδομένων μέσω "man-in-the-middle" επιθέσεων.

Για να συνδεθούμε σε έναν απομακρυσμένο υπολογιστή με SSH, χρησιμοποιούμε την εντολή ssh. Η σύνταξή της είναι απλή και έχει τη μορφή:

ssh username@remote\_host

* **username**: Είναι το όνομα χρήστη στον απομακρυσμένο υπολογιστή στον οποίο θέλεις να συνδεθείς. Αυτός ο χρήστης πρέπει να έχει δικαιώματα για απομακρυσμένη σύνδεση στο σύστημα.
* **remote\_host**: Είναι η διεύθυνση IP ή το όνομα του απομακρυσμένου υπολογιστή. Αυτή είναι η διεύθυνση του υπολογιστή στον οποίο επιθυμούμε να συνδεθούμε. Μπορεί να είναι μια στατική IP, μια δυναμική διεύθυνση ή ένα domain name.

Για παράδειγμα, αν θέλουμε να συνδεθούμε στον απομακρυσμένο υπολογιστή με IP 192.168.1.10 και όνομα χρήστη user1, η εντολή θα είναι:

ssh [user1@192.168.1.10](mailto:user1@192.168.1.10)

Αφού εκτελέσουμε την εντολή, θα μας ζητηθεί να εισάγουμε τον κωδικό πρόσβασης του χρήστη admin. Μόλις τον πληκτρολογήσουμε, θα συνδεθούμε στον απομακρυσμένο υπολογιστή και θα μπορούμε να εκτελέσουμε εντολές στο σύστημά του.

Ο **SSH daemon** (ή **sshd**) είναι το πρόγραμμα που εκτελείται στον απομακρυσμένο υπολογιστή και επιτρέπει τη σύνδεση μέσω του πρωτοκόλλου SSH. Ο δαίμονας αυτός ακούει συνεχώς για εισερχόμενες συνδέσεις στην προκαθορισμένη θύρα (συνήθως τη θύρα 22) και όταν εντοπίζει μια σύνδεση, ξεκινά τη διαδικασία αυθεντικοποίησης του χρήστη. Ο sshd επιτρέπει τη σύνδεση μέσω κωδικού πρόσβασης ή με χρήση κλειδιών (δημόσιου και ιδιωτικού), ενώ υποστηρίζει επίσης άλλες δυνατότητες ασφαλείας, όπως την επιβεβαίωση του απομακρυσμένου υπολογιστή μέσω κρυπτογραφικών αλγορίθμων. Επιπλέον, μπορεί να περιορίσει τις ενέργειες που μπορούν να εκτελούνται από τον συνδεδεμένο χρήστη, παρέχοντας έτσι έναν ασφαλή τρόπο απομακρυσμένης διαχείρισης συστημάτων. Ο δαίμονας sshd μπορεί να ρυθμιστεί μέσω του αρχείου διαμόρφωσης /etc/ssh/sshd\_config, το οποίο καθορίζει τις πολιτικές και τις ρυθμίσεις ασφαλείας για τις συνδέσεις SSH. Με την εντολή systemctl μπρούμε να ελέγξουμε την λειτουργία του sshd.

Στην περίπτωση που ο απομακρυσμένος υπολογιστής χρησιμοποιεί διαφορετική θύρα από την προεπιλεγμένη (η οποία είναι η θύρα 22), μπορείτε να ορίσετε την θύρα μέσω της παραμέτρου -p. Για παράδειγμα, αν η θύρα είναι η 2222, η εντολή θα είναι:

ssh -p 2222 admin@192.168.1.10

Αν, αντί για τον κωδικό πρόσβασης, χρειάζεται να χρησιμοποιήσεις ένα ιδιωτικό κλειδί (private key) για την πιστοποίηση, μπορείτε να το καθορίσετε με την παράμετρο -i. Το ιδιωτικό κλειδί συνήθως βρίσκεται σε έναν φάκελο όπως ~/.ssh/ και πρέπει να έχει τα σωστά δικαιώματα για να χρησιμοποιηθεί. Η εντολή με ιδιωτικό κλειδί θα είναι:

ssh -i /path/to/private\_key admin@192.168.1.10

Όπου το /path/to/private\_key είναι η διαδρομή του αρχείου του ιδιωτικού κλειδιού.

### Χρήση SSH για μεταφορά αρχείων

Εκτός από την απομακρυσμένη εκτέλεση εντολών, το SSH μπορεί να χρησιμοποιηθεί και για τη μεταφορά αρχείων μέσω πρωτοκόλλου SCP (Secure Copy Protocol) ή SFTP (Secure File Transfer Protocol). Η εντολή scp επιτρέπει την αντιγραφή αρχείων από τον τοπικό υπολογιστή στον απομακρυσμένο ή το αντίστροφο, ενώ η sftp επιτρέπει τη διαχείριση των αρχείων σε απομακρυσμένο σύστημα μέσω μιας ασφαλούς σύνδεσης.

Για παράδειγμα, για να αντιγράψουμε το αρχείο file.txt από τον τοπικό υπολογιστή στον απομακρυσμένο υπολογιστή, στον κατάλογο /path/to/destination, χρησιμοποιούμε την κάτωθι εντολή scp:

scp file.txt admin@192.168.1.10:/path/to/destination

### Συμπεράσματα

Η σύνδεση σε απομακρυσμένους υπολογιστές μέσω SSH είναι ένα ισχυρό και ασφαλές εργαλείο που χρησιμοποιείται ευρέως για την απομακρυσμένη διαχείριση συστημάτων. Είτε για την εκτέλεση εντολών, είτε για την ασφαλή μεταφορά αρχείων, το SSH παρέχει την ασφάλεια και την ευελιξία που απαιτείται σε περιβάλλοντα παραγωγής και διαχείρισης υπολογιστικών συστημάτων.

## Ρύθμιση χρήσης κλειδιών (δημόσιου-ιδιωτικού) για την πιστοποίηση της ταυτότητας του χρήστη χωρίς την εισαγωγή κωδικού.

Η χρήση κλειδιών (δημόσιου και ιδιωτικού) για την πιστοποίηση της ταυτότητας του χρήστη μέσω SSH επιτρέπει την ασφαλή σύνδεση σε απομακρυσμένους υπολογιστές χωρίς την ανάγκη για την εισαγωγή κωδικού πρόσβασης κάθε φορά. Αυτή η μέθοδος βασίζεται στην κρυπτογραφία και προσφέρει μεγαλύτερη ασφάλεια και ευκολία σε σύγκριση με τη χρήση παραδοσιακών κωδικών πρόσβασης.

### Δημιουργία ζεύγους κλειδιών

Η διαδικασία ξεκινά με τη δημιουργία ενός ζεύγους κλειδιών, το οποίο περιλαμβάνει δύο μέρη: το **ιδιωτικό κλειδί** και το **δημόσιο κλειδί**. Το ιδιωτικό κλειδί παραμένει στον τοπικό υπολογιστή και πρέπει να προστατεύεται με αυστηρά μέτρα ασφαλείας, ενώ το δημόσιο κλειδί μπορεί να διαμοιραστεί και να αποθηκευτεί στον απομακρυσμένο υπολογιστή με τον οποίο θέλουμε να συνδεθούμε.

Για να δημιουργήσουμε το ζεύγος κλειδιών, χρησιμοποιούμε την εντολή ssh-keygen στο τερματικό του τοπικού υπολογιστή.

Ακολουθεί η διαδικασία με βήματα:

1. Άνοιγμα του τερματικού και εκτέλεση της εντολής:

ssh-keygen

1. Θα μας ζητηθεί να καθορίσουμε τη διαδρομή για την αποθήκευση του κλειδιού. Η προεπιλεγμένη διαδρομή είναι ~/.ssh/id\_rsa. Εάν δεν θέλουμε να την αλλάξουμε, απλά πατάμε Enter.
2. Θα μας ζητηθεί επίσης να εισαγάγουμε έναν κωδικό πρόσβασης (passphrase) για το ιδιωτικό κλειδί. Αυτός ο κωδικός πρόσβασης παρέχει ένα επιπλέον επίπεδο ασφαλείας, αλλά είναι προαιρετικός. Αν δεν θέλουμε να χρησιμοποιήσουμε κωδικό πρόσβασης, απλά πατάμε Enter.

Μετά την εκτέλεση αυτών των βημάτων, δύο αρχεία θα δημιουργηθούν στον φάκελο ~/.ssh/:

* id\_rsa: Το ιδιωτικό κλειδί, το οποίο πρέπει να παραμείνει μυστικό και δεν πρέπει να το κοινοποιήσουμε.
* id\_rsa.pub: Το δημόσιο κλειδί, το οποίο μπορούμε να το μοιραστούμε με άλλους υπολογιστές για να μας επιτρέψουν να συνδεθούμε σε αυτούς χωρίς κωδικό πρόσβασης.

### Μεταφορά του δημοσίου κλειδιού στον απομακρυσμένο υπολογιστή

Αφού δημιουργήσουμε το ζεύγος κλειδιών, το επόμενο βήμα είναι να προσθέσουμε το δημόσιο κλειδί στον απομακρυσμένο υπολογιστή στον οποίον θέλουμε να συνδεθούμε. Για αυτόν τον σκοπό, υπάρχει η δυνατότητα να χρησιμοποιήσουμε την εντολή ssh-copy-id, η οποία αυτοματοποιεί την διαδικασία προσθήκης του δημόσιου κλειδιού στο αρχείο authorized\_keys στον απομακρυσμένο υπολογιστή.

Η εντολή για την προσθήκη του δημόσιου κλειδιού στον απομακρυσμένο υπολογιστή είναι η εξής:

ssh-copy-id username@remote\_host

Αντικαθιστούμε το username με το όνομα χρήστη στον απομακρυσμένο υπολογιστή και το remote\_host με την IP ή το domain name του απομακρυσμένου υπολογιστή. Αυτή η εντολή θα ζητήσει τον κωδικό πρόσβασης του χρήστη στον απομακρυσμένο υπολογιστή για να προσθέσει το δημόσιο κλειδί στο αρχείο ~/.ssh/authorized\_keys στον απομακρυσμένο υπολογιστή.

Εναλλακτικά, αν δεν έχουμε διαθέσιμο στο σύτημά μας το ssh-copy-id ή αν προτιμάμε να το κάνουμε χειροκίνητα, μπορούμε να αντιγράψουμε το περιεχόμενο του αρχείου id\_rsa.pub στον απομακρυσμένο υπολογιστή. Για να το κάνουμε αυτό, ανοίγουμε το αρχείο id\_rsa.pub στο τοπικό σύστημα και αντιγράφουμε το περιεχόμενο του. Στη συνέχεια, συνδεόμαστε σε απομακρυσμένο υπολογιστή μέσω SSH και επικολλάμε το περιεχόμενο στο τέλος του αρχείου ~/.ssh/authorized\_keys του απομακρυσμένου υπολογιστή. Αν το αρχείο δεν υπάρχει, πρέπει να το δημιουργήσουμε.

Αφού ολοκληρωθεί η προσθήκη του δημόσιου κλειδιού στον απομακρυσμένο υπολογιστή, μπορείς να συνδεθείς σε αυτόν χωρίς να απαιτείται η εισαγωγή του κωδικού πρόσβασης κάθε φορά. Η σύνδεση θα επιβεβαιώνεται μέσω της αντιστοίχισης του δημόσιου κλειδιού που έχεις αποθηκεύσει στον απομακρυσμένο υπολογιστή και του ιδιωτικού κλειδιού που έχεις στον τοπικό υπολογιστή σου.

Η εντολή για να συνδεθούμε είναι η εξής:

ssh username@remote\_host

Αν έχουμε ορίσει passphrase για το ιδιωτικό κλειδί, θα μας ζητηθεί να το εισαγάγουμε για να προχωρήσουμε στη σύνδεση.

### Ασφάλεια και πλεονεκτήματα

Η χρήση κλειδιών για την πιστοποίηση της ταυτότητας είναι πιο ασφαλής από την παραδοσιακή μέθοδο με κωδικό πρόσβασης. Κάποια πλεονεκτήματα της χρήσης κλειδιών είναι:

|  |  |
| --- | --- |
| **Ασφάλεια** | Τα κλειδιά SSH είναι εξαιρετικά δύσκολο να υποκλαπούν, ειδικά όταν χρησιμοποιούνται σε συνδυασμό με ένα ισχυρό passphrase. Σε αντίθεση με τους κωδικούς πρόσβασης, τα κλειδιά δεν μπορούν να μαντευτούν εύκολα από επιθέσεις brute force. |
| **Ευκολία** | Με την εγκατάσταση των κλειδιών, δεν χρειάζεται να εισαγλαγουμε κωδικούς πρόσβασης κάθε φορά που συνδεόμαστε, κάτι που διευκολύνει τη διαχείριση πολλαπλών συστημάτων ή συχνές απομακρυσμένες συνδέσεις. |
| **Αυτοματισμός** | Χρησιμοποιώντας κλειδιά, μπορούμε να ρυθμίσουμε απομακρυσμένες συνδέσεις ή αντιγραφές αρχείων χωρίς να απαιτείται ανθρώπινη παρέμβαση για την εισαγωγή κωδικών πρόσβασης, κάτι που είναι ιδιαίτερα χρήσιμο για αυτοματοποιημένες διαδικασίες. |

Συνολικά, η χρήση κλειδιών SSH προσφέρει μια ασφαλή, αξιόπιστη και βολική μέθοδο για τη σύνδεση σε απομακρυσμένα συστήματα χωρίς να απαιτείται η εισαγωγή κωδικών πρόσβασης.

## Περιορισμός της απευθείας σύνδεσης ως διαχειριστής και απενεργοποίηση του ελέγχου πιστοποίησης ταυτότητας με χρήση κωδικού

Ο περιορισμός της απευθείας σύνδεσης ως διαχειριστής (π.χ., root) και η απενεργοποίηση του ελέγχου πιστοποίησης ταυτότητας μέσω κωδικού πρόσβασης είναι κρίσιμα μέτρα ασφαλείας για την αποτροπή μη εξουσιοδοτημένων προσβάσεων σε έναν απομακρυσμένο υπολογιστή μέσω SSH. Αυτά τα μέτρα μειώνουν την επιφάνεια επίθεσης και αυξάνουν τη δυσκολία εκτέλεσης κακόβουλων ενεργειών, όπως brute-force επιθέσεις ή κατάχρηση δικαιωμάτων διαχειριστή.

### Περιορισμός σύνδεσης του διαχειριστή του συστήματος

Η σύνδεση ως διαχειριστής είναι επικίνδυνη, καθώς παρέχει απευθείας πρόσβαση σε όλα τα δικαιώματα του συστήματος. Για να περιορίσεις τη δυνατότητα απευθείας σύνδεσης ως root μέσω SSH:

1. **Επεξεργασία του αρχείου διαμόρφωσης του SSH daemon**: Ανοίγουμε το αρχείο /etc/ssh/sshd\_config με έναν επεξεργαστή κειμένου, π.χ.:

nano /etc/ssh/sshd\_config

1. **Εντοπισμός της παραμέτρου PermitRootLogin**: Εντοπίζουμε τη γραμμή που περιέχει την παράμετρο PermitRootLogin. Εάν η γραμμή είναι σχολιασμένη (προηγείται σύμβολο #), αφαιρούμε το σχόλιο και βεβαιωνόμαστε ότι η τιμή της παραμέτρου είναι:

PermitRootLogin no

Αυτό θα αποτρέψει τη δυνατότητα απευθείας σύνδεσης ως root.

1. **Αποθήκευση και επανεκκίνηση του SSH daemon**: Αποθηκεύουμε το αρχείο και επανεκκινούμε την υπηρεσία SSH για να εφαρμοστούν οι αλλαγές:

systemctl restart sshd.service

### Απενεργοποίηση πιστοποίησης μέσω κωδικού πρόσβασης

Η πιστοποίηση μέσω κωδικού είναι πιο ευάλωτη σε επιθέσεις brute-force και phishing, επομένως συνιστάται η χρήση ζεύγους κλειδιών (δημόσιου / ιδιωτικού) για την αυθεντικοποίηση. Για να απενεργοποιήσουμε την πιστοποίηση μέσω κωδικού ακολουθούμε τα παρακάτω βήματα:

1. **Επεξεργασία του αρχείου διαμόρφωσης**: Ανοίγουμε ξανά το αρχείο /etc/ssh/sshd\_config:

nano /etc/ssh/sshd\_config

1. **Εντοπισμός της παραμέτρου PasswordAuthentication**: Εντοπίζουμε τη γραμμή που περιέχει την παράμετρο PasswordAuthentication. Εάν η γραμμή είναι σχολιασμένη, αφαιρούμε το σχόλιο και ρυθμίζουμε την τιμή ως εξής:

PasswordAuthentication no

1. **Ενεργοποίηση της Πιστοποίησης με Κλειδιά**: Βεβαιωνόματε ότι η παράμετρος PubkeyAuthentication έχει οριστεί σε:

PubkeyAuthentication yes

Αυτό διασφαλίζει ότι η σύνδεση θα γίνεται αποκλειστικά με χρήση ζεύγους κλειδιών.

1. **Αποθήκευση και επανεκκίνηση του SSH daemon**: Αποθηκεύουμε τις αλλαγές και επανεκκινούμε την υπηρεσία SSH:

systemctl restart sshd.service

Μετά την εφαρμογή αυτών των αλλαγών, δοκιμάζουμε να συνδεθούμε στον απομακρυσμένο υπολογιστή με SSH, χρησιμοποιώντας έναν χρήστη με περιορισμένα δικαιώματα που έχει εξουσιοδοτηθεί μέσω ζεύγους κλειδιών. Στη συνέχεια, μπορούμε να αναβαθμίσουμε τα δικαιώματα του χρήστη σε root με την εντολή sudo, αν είναι απαραίτητο. Για να βεβαιωθούμε ότι οι αλλαγές λειτουργούν σωστά, μπορούμε να επιχειρήσουμε να συνδεθούμε ως root ή με κωδικό πρόσβασης, όπου θα πρέπει να απορριφθεί η σύνδεση.

### Οφέλη αυτών των μέτρων

|  |  |
| --- | --- |
| **Μείωση Κινδύνου** | Περιορίζει τις πιθανότητες επιτυχούς εκτέλεσης επιθέσεων brute-force. |
| **Αυξημένη Ασφάλεια** | Οι χρήστες συνδέονται μόνο μέσω κλειδιών, τα οποία είναι πολύ πιο ασφαλή από τους κωδικούς πρόσβασης. |
| **Περιορισμένη Πρόσβαση Διαχειριστή** | Ακόμα κι αν κάποιος αποκτήσει πρόσβαση σε έναν χρήστη με περιορισμένα δικαιώματα, θα χρειαστεί πρόσθετες εξουσιοδοτήσεις για να εκτελέσει κρίσιμες ενέργειες. |

Αυτά τα μέτρα, αν και προσθέτουν ένα επίπεδο πολυπλοκότητας στη διαμόρφωση, ενισχύουν σημαντικά την ασφάλεια του συστήματος και είναι ιδιαίτερα απαραίτητα για υπολογιστές που είναι προσβάσιμοι από το διαδίκτυο.

## Χρήση της υπηρεσίας fail2ban για την προστασία των συνδέσεων

Η υπηρεσία fail2ban είναι ένα ισχυρό εργαλείο ασφαλείας που προστατεύει τις συνδέσεις SSH και άλλες υπηρεσίες από κακόβουλες προσπάθειες πρόσβασης. Λειτουργεί παρακολουθώντας τα αρχεία καταγραφής (log files) του συστήματος και εντοπίζοντας αποτυχημένες προσπάθειες σύνδεσης ή άλλες ύποπτες δραστηριότητες. Όταν ανιχνεύσει έναν αριθμό αποτυχημένων προσπαθειών που ξεπερνά ένα καθορισμένο όριο, μπλοκάρει προσωρινά τη διεύθυνση IP του εισβολέα, αποτρέποντας περαιτέρω επιθέσεις

Για να εγκαταστήσουμε το fail2ban, εκτελούμε την εντολή:

apt update

apt install fail2ban

Σε διανομές που δεν χρησιμοποιούν apt, μπορούμε να χρησιμοποιήσουμε τον αντίστοιχο package manager (π.χ., yum για CentOS).

Ελέγχουμε αν η υπηρεσία έχει ξεκινήσει:

systemctl status fail2ban

Και τέλος βεβαιώνόμαστε ότι η υπηρεσία ρυθμίζεται να ξεκινά αυτόματα κατά την εκκίνηση:

systemctl enable fail2ban

### Βασικές ρυθμίσεις

Το fail2ban χρησιμοποιεί αρχεία διαμόρφωσης για να καθορίσει ποιες υπηρεσίες θα προστατεύει και πώς θα ανταποκρίνεται στις ύποπτες δραστηριότητες.

Το κύριο αρχείο διαμόρφωσης είναι το /etc/fail2ban/jail.conf. Ωστόσο, για να αποφεύγονται προβλήματα κατά την αναβάθμιση της εφαρμογής, συνιστάται να δημιουργήσουμε το /etc/fail2ban/jail.local:

cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

Το αρχείο jail.conf διαβάζεται πρώτο από το σύστημα και στη συνέχεια διαβάζεται το αρχείο jail.local, με τις ρυθμίσεις του τελευταίου να υπερισχύουν αυτών που ευρίσκονται και στο jail.conf. Έτσι, το αρχείο jail.local δεν χρειάζεται να περιλαμβάνει όλες τις ρυθμίσεις του jail.conf, παρά μόνο εκείνες που θέλουμε να παρακάμψουμε. Συνεπώς, οι τροποποιήσεις πρέπει να γίνονται στο αρχείο jail.local και όχι στο jail.conf, ώστε να αποφεύγονται προβλήματα συγχώνευσης κατά την αναβάθμιση.

Ανοίγουμε το αρχείο jail.local με καποιον επεξεργαστή κειμένου και μεταβαίνουμε στην ενότητα [DEFAULT], όπου μπορούμε να ορίσουμε τις κάτωθι βασικές παραμέτρους:

* + bantime: Ορίζει τη διάρκεια αποκλεισμού μιας IP (π.χ., 10m για 10 λεπτά ή 1h για μία ώρα). Εάν θέλουμε να είναι μόνιμος ο αποκλεισμός, χρησιμοποιούμε την τιμή -1.
  + findtime: Ορίζει το χρονικό διάστημα στο οποίο πρέπει να καταγραφεί ένας συγκεκριμένος αριθμός αποτυχημένων προσπαθειών για να ενεργοποιηθεί ο αποκλεισμός (π.χ., 10m).
  + maxretry: Ορίζει τον αριθμό των αποτυχημένων προσπαθειών σύνδεσης που απαιτούνται για να ενεργοποιηθεί ο αποκλεισμός.

Παράδειγμα:

[DEFAULT]

bantime = 10m

findtime = 10m

maxretry = 5

Στη συνέχεια επιλέγουμε να ρυθμίσουμε το fail2ban για την υπηρεσία sshd. Προσθέτουμε τα κάτωθι στην ενότητα [sshd]:

[sshd]

enabled = true

port = ssh

logpath = /var/log/auth.log

maxretry = 5

όπου:

* + enabled: Ενεργοποιεί την προστασία για την υπηρεσία sshd.
  + logpath: Καθορίζει το αρχείο καταγραφής που παρακολουθεί το fail2ban για προσπάθειες σύνδεσης.

Αποθηκεύουμε το αρχείο jail.local και για να εφαρμόσουμε τις αλλαγές επανεκκινούμε την υπηρεσία:

systemctl restart fail2ban.service

### Έλεγχος της κατάστασης

Μπορούμε να ελέγξουμε ποια jails είναι ενεργά με την κάτωθι εντολή:

fail2ban-client status

η οποία μας πληροφορεί ότι υπάρχει ενεργό ένα jail:

Status

|- Number of jail: 1

`- Jail list: sshd

Για να πάρουμε συγκεκριμένες πληροφορίες για το jail της υπηρεσίας sshd:

fail2ban-client status sshd

όπου με το κάτωθι αποτέλεσμα:

Status for the jail: sshd

|- Filter

| |- Currently failed: 1

| |- Total failed: 23829

| `- File list: /var/log/auth.log

`- Actions

|- Currently banned: 2

|- Total banned: 2569

`- Banned IP list: 192.0.2.1 198.51.100.1

μας ειδοποιεί ότι υπάρχουν 2 αποκλεισμένες IP διευθύνσεις. Για να δούμε μόνο τις αποκλεισμένες IP διευθύνσεις δίνουμε:

fail2ban-client banned

με αποτέλεσμα:

[{'sshd': ['192.0.2.1', '198.51.100.1']}]

Επίσης, μπορούμε να προσθέσουμε χειροκίνητα μία IP διεύθυνση στη λίστα αποκλεισμένων IP διευθύνσεων ως εξής:

fail2ban-client set sshd banip <IP>

Η κατάργηση του αποκλεισμού μιας IP διεύθυνσης επιτυγχάνεται ως εξής:

fail2ban-client set sshd unbanip <IP>

### Πλεονεκτήματα

|  |  |
| --- | --- |
| **Αυτόματη Απάντηση** | Αναγνωρίζει και αποκλείει ύποπτες δραστηριότητες αυτόματα, μειώνοντας τον χρόνο αντίδρασης. |
| **Ευελιξία** | Μπορεί να προσαρμοστεί για την προστασία πολλών διαφορετικών υπηρεσιών πέρα από το SSH, όπως το FTP ή ο Apache. |
| **Αύξηση Ασφάλειας** | Μειώνει τις πιθανότητες επιτυχημένων brute-force επιθέσεων. |
| **Εύκολη Διαχείριση** | Παρέχει απλές εντολές για την παρακολούθηση και τη διαχείριση αποκλεισμένων διευθύνσεων IP. |

Το fail2ban είναι απαραίτητο για την ασφάλεια οποιουδήποτε συστήματος είναι προσβάσιμο μέσω SSH, ειδικά όταν εκτίθεται στο διαδίκτυο. Η σωστή διαμόρφωσή του μπορεί να προστατεύσει αποτελεσματικά το σύστημα από πολλές κοινές απειλές.

## Άσκηση – Ρύθμιση και ασφάλιση του SSH

**Στόχος**: Να κατανοήσετε τις βασικές έννοιες της ρύθμισης και της ασφάλισης του πρωτοκόλλου SSH, να εφαρμόσετε πρακτικά μέτρα για την προστασία των απομακρυσμένων συνδέσεων και να εξασφαλίσετε ασφαλή πρόσβαση σε έναν απομακρυσμένο υπολογιστή.

**Προσδοκώμενα αποτελέσματα**: Η σύνδεση SSH θα επιτρέπεται μόνο μέσω κλειδιών, και όχι με κωδικούς πρόσβασης. Ο χρήστης root δεν θα μπορεί να συνδεθεί απευθείας.

* 1. Ρυθμίστε την κατάλληλη παράμετρο στο etc/ssh/sshd\_config για να αποτρέψετε τη σύνδεση ως root:
  2. Ενεργοποιήστε τη σύνδεση SSH στο etc/ssh/sshd\_config μόνο μέσω κλειδιών:

#### Για να εξασφαλίσετε τη σύνδεση με κλειδιά, δημιουργήστε τοζεύγος δημόσιου και ιδιωτικού κλειδιού:

* 1. Αντιγράψτε το δημόσιο κλειδί στον απομακρυσμένο υπολογιστή remote\_host ως χρήστης user1:

### Ενδεικτική επίλυση

* 1. Ρυθμίστε την κατάλληλη παράμετρο στο etc/ssh/sshd\_config για να αποτρέψετε τη σύνδεση ως root:

PermitRootLogin no

* 1. Ενεργοποιήστε τη σύνδεση SSH στο etc/ssh/sshd\_config μόνο μέσω κλειδιών:

PasswordAuthentication no

PubkeyAuthentication yes

#### Για να εξασφαλίσετε τη σύνδεση με κλειδιά, δημιουργήστε τοζεύγος δημόσιου και ιδιωτικού κλειδιού:

ssh-keygen -t rsa -b 4096

* 1. Αντιγράψτε το δημόσιο κλειδί στον απομακρυσμένο υπολογιστή remote\_host ως χρήστης user1:

ssh-copy-id user1@remote\_host

## Aρχιτεκτονική καταγραφής συμβάντων με το journald

Το journald είναι το σύστημα καταγραφής (logging) στο systemd, το οποίο συλλέγει και αποθηκεύει μηνύματα καταγραφής από τον πυρήνα, τις υπηρεσίες, και τις εφαρμογές του συστήματος. To journald διαχειρίζεται όλα τα συμβάντα που σχετίζονται με το σύστημα, επιτρέποντας στον διαχειριστή να έχει εύκολη πρόσβαση σε αναλυτικές πληροφορίες.

Η αρχιτεκτονική καταγραφής συμβάντων του journald αποτελεί ένα από τα βασικά υποσυστήματα του systemd για τη συλλογή και διαχείριση των logs στο Linux, με κύρια αποστολή την αποδοτική συγκέντρωση και αποθήκευση πληροφοριών για τα συμβάντα που λαμβάνουν χώρα στο σύστημα. Ακολουθεί αναλυτική περιγραφή των βασικών στοιχείων της αρχιτεκτονικής του:

Το journald συλλέγει συμβάντα από πολλαπλές πηγές όπως:

|  |  |
| --- | --- |
| **Systemd services** | Καταγράφει τα logs από δαίμονες και υπηρεσίες που διαχειρίζεται το systemd. |
| **Syslog API** | Λαμβάνει μηνύματα από εφαρμογές και υπηρεσίες που χρησιμοποιούν το παραδοσιακό πρωτόκολλο syslog. |
| **Standard input (stdin), output (stdout), και error (stderr)** | Συγκεντρώνει την έξοδο των υπηρεσιών που χρησιμοποιούν αυτά τα streams. |
| **Kernel messages** | Λαμβάνει μηνύματα από τον πυρήνα του Linux, μέσω του kmsg, για βασικά συστήματα μηνύματα και σφάλματα. |

Το journald αποθηκεύει τα δεδομένα σε δυαδική μορφή για μεγαλύτερη απόδοση και καλύτερη συμπίεση. Τα μηνύματα καταγραφής μορφοποιούνται με πρόσθετα μεταδεδομένα (π.χ., χρόνος, υπηρεσία, χρήστης, PID) που βοηθούν στην αναζήτηση και ανάλυση. Κάθε καταγραφή είναι αυτοτελής, κάτι που επιτρέπει την ευκολότερη ανάλυση, καθώς η πληροφορία είναι πλήρης χωρίς να χρειάζεται εξωτερική συσχέτιση δεδομένων.

Τα logs αποθηκεύονται σε στο /var/log/journal, ή στο /run/log/journal για προσωρινή καταγραφή. Οι ρυθμίσεις διάρκειας ζωής και μεγέθους των logs μπορούν να καθοριστούν, και η αποθήκευση να γίνεται είτε προσωρινά στη RAM είτε μόνιμα στο δίσκο. Το journald μπορεί να συμπιέζει αυτόματα και να αρχειοθετεί logs, ενώ υποστηρίζει και τον περιορισμό μεγέθους αρχείου, ώστε να μην γεμίζει ο αποθηκευτικός χώρος.

Με την εντολή journalctl, οι διαχειριστές μπορούν να αναζητούν και να φιλτράρουν τις καταγραφές με κριτήρια όπως ο δαίμονας (-u service\_name), ο χρόνος (--since, --until), και το επίπεδο σφάλματος (-p).

Η πρόσβαση στα logs μπορεί να περιοριστεί για συγκεκριμένους χρήστες ή ομάδες για λόγους ασφαλείας, και τα δεδομένα μπορούν να κρυπτογραφηθούν.Οι χρήστες που ανήκουν στην ομάδα systemd-journal μπορούν να έχουν ανάγνωση των καταγραφών, χωρίς πλήρη δικαιώματα διαχειριστή.

Αυτή η δομή καθιστά το journald εξαιρετικά αποδοτικό και ασφαλές, παρέχοντας μια ολοκληρωμένη λύση για την παρακολούθηση, ανάλυση, και διαχείριση των συστημάτων logs σε περιβάλλοντα Linux.

Για να ξεκινήσουμε ή για να σταματήσουμε το journald χρησιμοπούμε τις εντολές systemctl start systemd-journald.service και systemctl stop systemd-journald.service αντίστοιχα. Ο δαίμονας είναι συνήθως ενεργοποιημένος από προεπιλογή, καθώς είναι κρίσιμος για την καταγραφή συμβάντων στο σύστημα. Δεν συνιστάται να τον απενεργοποιούμε, εκτός αν έχουμε συγκεκριμένο λόγο και εναλλακτικό σύστημα καταγραφής.

Τα βασικά χαρακτηριστικά του journald είναι τα κάτωθι:

|  |  |
| --- | --- |
| **Κεντρική αποθήκευση** | Συλλέγει δεδομένα από πολλές πηγές σε μια κοινή βάση, αποθηκεύοντας τα μηνύματα καταγραφής σε δυαδικά αρχεία (journal files). |
| **Δυαδική αποθήκευση** | Αντί για απλό κείμενο, χρησιμοποιεί δυαδική αποθήκευση, ώστε να βελτιώσει την αποδοτικότητα και να προσφέρει δυνατότητες όπως indexing, αναζήτηση και φίλτρα. |
| **Ευέλικτο σύστημα αναζήτησης** | Οι εγγραφές καταγραφής είναι εύκολα αναζητήσιμες με φίλτρα όπως η υπηρεσία, ο τύπος συμβάντος και το ID της διαδικασίας. |
| **Διατήρηση ιστορικού και αρχειοθέτηση** | Υποστηρίζει διαχείριση μεγέθους, διαγραφή παλιών δεδομένων και αρχειοθέτηση. |

## Εύρεση και ερμηνεία καταγραφών για την επίλυση προβλημάτων

Η εύρεση και η ερμηνεία των καταγραφών είναι κρίσιμες διαδικασίες για την επίλυση προβλημάτων και την ανάλυση της λειτουργίας του συστήματος στο Linux. Το journald, σε συνδυασμό με την εντολή journalctl, προσφέρει ισχυρά εργαλεία για να βρίσκουμε συγκεκριμένα σφάλματα, να φιλτράρουμε καταγραφές, και να κατανοούμε την αιτία των προβλημάτων σε υπηρεσίες και συστήματα.

Μερικές βασικές χρήσεις της εντολής journalctl έιναι οι παρακάτω:

1. **Προβολή καταγραφών από συγκεκριμένη υπηρεσία**: Για να δούμε τα logs μιας συγκεκριμένης υπηρεσίας, όπως η sshd, γράφουμε:

journalctl -u sshd.service

1. **Φιλτράρισμα βάσει χρονικής περιόδου**: Για να δούμε καταγραφές για μια συγκεκριμένη χρονική περίοδο:

journalctl --since "2024-10-30" --until "2024-11-01"

η οποία ενδεικτικά έχει ως αποτέλεσμα την προβολή των κάτωθι καταγραφών:

Oct 10 01:42:30 ubu01 sshd[2234530]: Disconnected from invalid user kvz 93.183.70.38 port 48780 [preauth]

Oct 10 01:42:32 ubu01 sshd[2234535]: Invalid user mberberi from 64.225.11.226 port 32870

Oct 10 01:42:33 ubu01 sshd[2234535]: Received disconnect from 64.225.11.226 port 32870:11: Bye Bye [preauth]

Oct 10 01:42:33 ubu01 sshd[2234535]: Disconnected from invalid user mberberi 64.225.11.226 port 32870 [preauth]

Oct 10 01:42:37 ubu01 sshd[2234541]: Invalid user minecraftserver from 185.100.53.71 port 48692

Oct 10 01:42:37 ubu01 sshd[2234541]: Received disconnect from 185.100.53.71 port 48692:11: Bye Bye [preauth]

Oct 10 01:42:37 ubu01 sshd[2234541]: Disconnected from invalid user minecraftserver 185.100.53.71 port 48692 [preauth]

Oct 10 01:42:42 ubu01 sshd[2234547]: Invalid user luzhp from 45.5.159.36 port 35486

Oct 10 01:42:42 ubu01 sshd[2234547]: Received disconnect from 45.5.159.36 port 35486:11: Bye Bye [preauth]

Oct 10 01:42:42 ubu01 sshd[2234547]: Disconnected from invalid user luzhp 45.5.159.36 port 35486 [preauth]

Oct 10 01:42:54 ubu01 sshd[2234563]: Invalid user luzhp from 181.62.251.76 port 39090

SHA256:QVYNM9aNGndxO5lJ5nH36QTiZ9RpEi5Yo+GCGmaOfHo

Oct 10 01:42:54 ubu01 sshd[2234565]: pam\_unix(sshd:session): session opened for user dimkar by (uid=0)

Oct 10 01:42:54 ubu01 sshd[2234563]: Received disconnect from 181.62.251.76 port 39090:11: Bye Bye [preauth]

Oct 10 01:42:54 ubu01 sshd[2234563]: Disconnected from invalid user luzhp 181.62.251.76 port 39090 [preauth]

Για να δούμε τις καταγραφές των τελευταίων 2 ωρών:

journalctl --since "2 hours ago"

1. **Καταγραφή πυρήνα**: Για να δούμε καταγραφές που προέρχονται από τον πυρήνα:

journalctl -k

1. **Αναζήτηση με βάση τη σοβαρότητα**: Οι καταγραφές μπορούν να φιλτραριστούν βάσει του επιπέδου σοβαρότητας. Για να δούμε τα κρίσιμα σφάλματα αλλά και τα logs σοβαρότερων επιπέδων δίνουμε την εντολή:

journalctl -p crit

Τα επίπεδα σοβαρότητας περιλαμβάνουν τα εξής:

|  |  |
| --- | --- |
| **emerg** | Άμεσος κίνδυνος. |
| **alert** | Ενέργεια απαραίτητη. |
| **crit** | Κρίσιμο σφάλμα. |
| **err** | Σφάλμα. |
| **warning** | Προειδοποίηση. |
| **notice, info, debug** | Λιγότερο σοβαρές καταστάσεις. |

1. **Παρακολούθηση καταγραφών σε πραγματικό χρόνο**:

Για να δούμε τα logs σε πραγματικό χρόνο, μπορούμε να χρησιμοποιήσουμε το -f:

journalctl -f

Ας δούμε ένα παράδειγμα με τη χρήση του journalctl για να προβάλουμε τα σφάλματα του httpd (Apache) σε πραγματικό χρόνο:

journalctl -u httpd.service -p err -f

Το journald προσφέρει εντολές για διαχείριση χώρου, εφόσον τα log αρχεία μπορεί να γίνουν πολύ μεγάλα. Μπορούμε να καθαρίσουμε τα αρχεία του journal με τις παρακάτω εντολές:

1. **Καθαρισμός με όριο μεγέθους**: Για να μειώσουμε τα logs και να αφήσουμε μόνο τα τελευταία 500 MB:

journalctl --vacuum-size=500M

1. **Καθαρισμός με χρονικό όριο**: Για να διαγράψουμε καταγραφές παλαιότερες των 30 ημερών:

journalctl --vacuum-time=30d

1. **Καθαρισμός με όριο αρχείων**: Για να αφήσουμε μόνο τα τελευταία 10 αρχεία καταγραφής:

journalctl --vacuum-files=10

1. **Καθαρισμός του αριθμού καταγραφών**:

journalctl -n x

όπου το x αντιτοιχεί στον επιθμητό αριθμό καταγραφών που θέλουμε να εμφανίσουμε.

Εφόσον χρειαστεί, μπορούμε να εξαγάγουμε συγκεκριμένα σύνολα καταγραφών σε ένα αρχείο για αναλυτικότερη επεξεργασία ή αποστολή:

journalctl -u sshd.service --since "2023-10-01" > sshd\_logs\_oct\_1.txt

Με αυτές τις μεθόδους, η αναζήτηση και η ερμηνεία των καταγραφών γίνεται αποτελεσματική, βοηθώντας στην ταχεία διάγνωση και επίλυση προβλημάτων στο σύστημα και στις υπηρεσίες του.

Εν κατακλείδι, τα βασικά πλεονεκτήματα του journald είναι:

|  |  |
| --- | --- |
| **Συγκέντρωση πληροφοριών** | Καταγράφει όλα τα μηνύματα σε ένα ενιαίο μέρος, επιτρέποντας εύκολη πρόσβαση και αναζήτηση. |
| **Δομημένη αποθήκευση** | Η δυαδική μορφή προσφέρει δυνατότητες όπως η γρήγορη αναζήτηση και η συμπίεση. |
| **Ασφάλεια** | Προσφέρει έλεγχο πρόσβασης μέσω πολιτικών (ACLs) που καθορίζουν ποιοι χρήστες μπορούν να δουν τα logs. |
| **Ευκολία στην παρακολούθηση** | Δίνει δυνατότητα παρακολούθησης σε πραγματικό χρόνο για τα logs, κάτι πολύ χρήσιμο για debugging. |

Το journald αποτελεί ένα ισχυρό εργαλείο για τη διαχείριση των καταγραφών στο systemd, και σε συνδυασμό με την εντολή journalctl επιτρέπει στον διαχειριστή να αναζητά, να φιλτράρει και να διαχειρίζεται τις καταγραφές αποτελεσματικά.

## Διατήρηση των εγγραφών συμβάντων

To journald αποθηκεύει τις εγγραφές στο /var/log/journal, εάν έχει δημιουργηθεί αυτός ο φάκελος και οι σχετικές ρυθμίσεις έχουν επιτραπεί. *Διαφορετικά, οι εγγραφές αποθηκεύονται μόνο προσωρινά στη μνήμη και χάνονται μετά από επανεκκίνηση*.

Το αρχείο ρυθμίσεων του journald βρίσκεται στο /etc/systemd/journald.conf, και οι κύριες ρυθμίσεις περιλαμβάνουν:

|  |  |
| --- | --- |
| **Storage** | Καθορίζει πού θα αποθηκεύονται τα αρχεία του journal (persistent, volatile, auto, ή none). |
| **Compress** | Αν είναι ενεργό, τα δεδομένα θα συμπιέζονται για εξοικονόμηση χώρου. |
| S**ystemMaxUse** και **SystemMaxFileSize** | Καθορίζουν το μέγιστο μέγεθος που μπορεί να καταλάβει το journal στο σύστημα. |

Παραθέτουμε παρακάτω ένα ενδεικτικό αρχείο ρυθμίσεων:

[Journal]

# Καθορίζει πού αποθηκεύονται οι εγγραφές του journald.

# Η τιμή "persistent" δημιουργεί έναν μόνιμο κατάλογο για τη διατήρηση εγγραφών.

Storage=persistent

# Καθορίζει το μέγιστο μέγεθος των αρχείων καταγραφής.

# Αλλαγή της τιμής ανάλογα με τις ανάγκες σας.

SystemMaxUse=1G

# Καθορίζει το μέγιστο μέγεθος ενός αρχείου καταγραφής πριν δημιουργηθεί νέο.

SystemMaxFileSize=200M

# Καθορίζει το χρονικό διάστημα διατήρησης των αρχείων καταγραφής.

# Εδώ παραμένουν για 2 εβδομάδες.

MaxRetentionSec=2week

# Ενεργοποίηση της συμπίεσης αρχείων καταγραφής για εξοικονόμηση χώρου.

Compress=yes

# Καθορίζει αν θα απορρίπτονται καταγραφές όταν δεν υπάρχει αρκετός χώρος.

SystemKeepFree=50M

Αφού κάνετε τις αλλαγές στο αρχείο journald.conf, θα πρέπει να επανεκκινήστε τον δάιμονα:

systemctl restart systemd-journald.service

Μπορούμε να ελέγξουμε την τρέχουσα διαμόρφωση με την κάτωθι εντολή:

journalctl --disk-usage

## Ρύθμιση συγχρονισμού ώρας και ζώνης ώρας με το NTP

Η διατήρηση του συγχρονισμού της ώρας και η σωστή ρύθμιση της ζώνης ώρας είναι κρίσιμες για την ακρίβεια των timestamps στα αρχεία καταγραφής. Παρακάτω περιγράφονται τα απαραίτητα βήματα.

Η σωστή ρύθμιση της ζώνης ώρας διασφαλίζει ότι τα timestamps των αρχείων καταγραφής αντιστοιχούν στην τοπική ώρα. Μπορούμε να ελέγξουμε την τρέχουσα ζώνης ώρας με την εντολή timedatectl. Η έξοδος θα δείχνει τη ζώνη ώρας, την τοπική ώρα και την ώρα UTC.

Local time: Fri 2024-12-13 09:00:21 EET

Universal time: Fri 2024-12-13 07:00:21 UTC

RTC time: Fri 2024-12-13 07:00:21

Time zone: Europe/Athens (EET, +0200)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

Για να ρυθμίσουμε μια συγκεκριμένη ζώνη ώρας (πχ. ζώνη ώρας Ελλάδος) μπορούμε να το πετύχουμε με την εντολή:

timedatectl set-timezone Europe/Athens

Αν εκτελέσουμε ξανά την timedatectl μπορούμε να επιβεβαιώσουμε τη ρύθμιση.

Το NTP διασφαλίζει ότι το σύστημα παραμένει συγχρονισμένο με αξιόπιστους διακομιστές ώρας. Για την λειτουργία του απαιτείται να τρέχει ο δάιμονας systemd-timesyncd. Για να ξεκινήσουμε ή για να σταματήσουμε το systemd-timesyncd χρησιμοπούμε τις εντολές systemctl start systemd-timesyncd.service και systemctl stop systemd-timesyncd.service αντίστοιχα.

Αρχικά βεβαιωνόμαστε ότι η ώρα του συστήματος είναι σωστή με την εντολή date:

Fri 13 Dec 2024 09:10:54 AM EET

όπου το EET έιναι η ζώνη ώρας ανατολικής Ευρώπης, Eastern European Time, η οποία βρίσκεται δύο ώρες μπροστά από τη UTC και χρησιμοποιείται σε χώρες της Ευρώπης, της Βόρειας Αφρικής και της Μέσης Ανατολής.

Για να εξασφαλίσουμε ότι τα καταγεγραμμένα συμβάντα έχουν τη σωστή ώρα, δίνουμε την εντολή:

journalctl

Τα περισσότερα συστήματα καταγράφουν τα συμβάντα στην τοπική ώρα από προεπιλογή. Αν θέλουμε να δούμε τα αρχεία καταγραφής σε UTC τότε χρησιμοπούμε το flag --utc στην εντολή journalctl:

journalctl --utc

Με αυτές τις ρυθμίσεις, το σύστημά θα παραμένει συγχρονισμένο μέσω NTP, και η ζώνη ώρας θα διασφαλίζει ότι τα timestamps στα αρχεία καταγραφής θα είναι ακριβή.

Η εντολή timedatectl set-time χρησιμοποιείται για τη χειροκίνητη ρύθμιση της ημερομηνίας και της ώρας του συστήματος. Η σύνταξή της απαιτεί την παροχή της ημερομηνίας και της ώρας στην μορφή YYYY-MM-DD HH:MM:SS, όπου YYYY είναι το έτος, MM ο μήνας, DD η ημέρα, και ακολουθεί η ώρα σε μορφή 24ώρου. Για παράδειγμα, η εντολή

timedatectl set-time "2024-12-16 14:30:00"

ρυθμίζει το σύστημα στην 16η Δεκεμβρίου 2024 και ώρα 14:30. Αυτή η εντολή είναι χρήσιμη σε περιπτώσεις όπου ο συγχρονισμός NTP δεν είναι διαθέσιμος ή χρειάζεται προσωρινά να οριστεί μια συγκεκριμένη ώρα.

## Ασκηση - Επισκόπηση, διατήρηση του συστήματος καταγραφής (system journal) και ρύθμιση της ώρας συστήματος μέσω του πρωτοκόλλου NTP

Η άσκηση αναφέρεται στη διατήρηση του συστήματος καταγραφής (system journal) και τη ρύθμιση της ώρας συστήματος μέσω του NTP (Network Time Protocol) για την εν γένει καλή λειτουργία και συγχρονισμό του συστήματος.

**Σκοπός:** Ο σκοπός της άσκησης είναι να κατανοήσετε και να εφαρμόσετε τις διαδικασίες για τη διατήρηση του συστήματος καταγραφής (system journal) και τη ρύθμιση της ώρας συστήματος μέσω του πρωτοκόλλου NTP.

**Προσδoκώμενα αποτελέσματα:** Με την ολοκλήρωση της άσκησης, θα έχετε αποκτήσει πρακτικές γνώσεις για την εξασφάλιση ότι η ώρα του συστήματος παραμένει συγχρονισμένη και ότι τα δεδομένα καταγραφής είναι προσβάσιμα και διαχειρίσιμα, κάτι που είναι κρίσιμο για τη διαχείριση συστημάτων σε περιβάλλοντα παραγωγής.

* + 1. **Γενική Επισκόπηση των καταγραφών (logs):**
    2. **Επισκόπηση μόνο των τελευταίων 3 καταγραφών που αφορούν το SSH:**
    3. ‘Ελεγχος της κατάστασης της υπηρεσίας NTP:
    4. Ρύθμιση της ώρας χειροκίνητα @15.30:

### Ενδεικτική επίλυση

* 1. **Γενική Επισκόπηση των καταγραφών (logs):**

journalctl

* 1. **Επισκόπηση μόνο των τελευταίων 3 καταγραφών που αφορούν το SSH:**

journalctl -u ssh -n 3

* 1. ‘Ελεγχος της κατάστασης της υπηρεσίας NTP:

systemctl status systemd-timesyncd.service

* 1. Ρύθμιση της ώρας χειροκίνητα @15.30:

timedatectl set-time '2024-12-18 15:30:00'

## Ερωτήσεις αυτοαξιολόγησης

1. Ποια είναι η εντολή για την επανεκκίνηση της υπηρεσίας SSH;  
   Α. sudo restart ssh-service  
   Β. systemctl restart ssh  
   Γ. ssh restart  
   Δ. service sshd restart
2. Ποιος είναι ο σκοπός του αρχείου /etc/ssh/sshd\_config;  
   Α. Να αποθηκεύει τα δημόσια κλειδιά των χρηστών.  
   Β. Να ρυθμίζει τις παραμέτρους του SSH Server.  
   Γ. Να καταγράφει τα συμβάντα του SSH.  
   Δ. Να περιέχει τις λίστες αποκλεισμένων IP.
3. Ποια επιλογή στο αρχείο ρυθμίσεων SSH αποτρέπει τη σύνδεση ως χρήστης root;  
   Α. RootLogin off  
   Β. AllowRootLogin false  
   Γ. PermitRootLogin no  
   Δ. DenyRootAccess true
4. Ποια είναι η εντολή για να αντιγράψετε το δημόσιο κλειδί σε έναν απομακρυσμένο υπολογιστή;  
   Α. scp id\_rsa.pub user@host:/authorized\_keys  
   Β. ssh-copy-id user@host  
   Γ. ssh-keysend user@host  
   Δ. scp -i id\_rsa.pub user@host:/etc/ssh
5. Ποιος μηχανισμός μπορεί να χρησιμοποιηθεί για την αποτροπή brute-force επιθέσεων στο SSH;  
   Α. iptables  
   Β. fail2ban  
   Γ. ufw  
   Δ. ssh-keygen
6. Ποια ρύθμιση επιτρέπει μόνο τη σύνδεση μέσω κλειδιών στο SSH;  
   Α. PasswordAuthentication no  
   Β. KeyOnlyAuthentication true  
   Γ. UseOnlyKeys yes  
   Δ. PubkeyOnly yes
7. Ποια εντολή χρησιμοποιούμε για να δούμε όλα τα logs από το journald;

**A. journalctl**

B**.** journald status

**Γ.** systemctl log

**Δ.** systemctl journal

1. Πώς μπορούμε να δούμε τα τελευταία 50 συμβάντα με το journalctl;  
   **A. journalctl -n 50  
   B.** journalctl --last 50  
   **Γ**. journalctl --tail 50  
   **Δ**. journalctl --recent 50
2. Πώς μπορούμε να φιλτράρουμε τις καταγραφές του journalctl για ένα συγκεκριμένο χρονικό διάστημα, π.χ., από την 1η Ιανουαρίου 2024;  
   **Α**. journalctl --since "2024-01-01"  
   **Β**. journalctl --date "2024-01-01"  
   **Γ**. journalctl -s "2024-01-01"  
   **Δ.** journalctl --filter-date "2024-01-01"
3. Ποια επιλογή του journalctl χρησιμοποιείται για την εμφάνιση των καταγεγραμμένων μηνυμάτων από το επίπεδο προειδοποίησης (warning) συμπεριλαμβανομένων και των μηνυμάτων σοβαρότερων επιπέδων;  
   **Α**. journalctl --priority warning  
   **Β**. journalctl -p warning  
   **Γ**. journalctl --level warn  
   Δ. journalctl -l warning

### Απαντήσεις

* + - 1. Β
      2. Β
      3. Γ
      4. Β
      5. Β
      6. Α
      7. Α
      8. Α
      9. Α
      10. Β

# Διαχείριση Δικτύων

Οι περισσότερες λειτουργίες ενός εξυπηρετητή εμπλέκουν άμεσα ή έμμεσα δικτυακές λειτουργίες. Δεν είναι και πολύ χρήσιμος ένας εξυπηρετητής όπου για να προσφέρει κάποια υπηρεσία πρέπει να έρθει κανείς σε επαφή με το φυσικό μηχάνημα και που δεν μπορούν να έχουν πρόσβαση στις υπηρεσίες του άλλοι υπολογιστές.

Η γνώση βασικών εννοιών δικτύων και η ικανότητα σωστής δικτυακής ρύθμισης είναι απαραίτητα εφόδια. Σε αυτό το κεφάλαιο θα περιγραφούν θεμελιώδεις έννοιες στη διευθυνσιοδότηση δικτύου και τη δρομολόγηση για έναν εξυπηρετητή. Κατόπιν θα γίνουν δοκιμές και έλεγχοι τις τρέχουσες ρυθμίσεις δικτύου μέσω της γραμμής εντολών, καθώς και διαχείριση των ρυθμίσεων δικτύου και των συσκευές με χρήση του **nmcli**. Στη συνέχεια θα επιχειρηθεί αλλαγή των ρυθμίσεων δικτύου μέσω της επεξεργασίας των αρχείων ρυθμίσεων, καθώς και η ρύθμιση του στατικού ονόματος ενός εξυπηρετητή, η επίλυση ονομάτων (name resolution) ο έλεγχος των αποτελεσμάτων.

## Βασικές έννοιες δικτύων

### Το δικτυακό μοντέλο TCP/IP

Το TCP/IP είναι ένα αφαιρετικό δικτυακό μοντέλο που αποτελείται από τέσσερα επίπεδα. Περιγράφει πώς διαλειτουργούν διάφορα πρωτόκολλα, έτσι ώστε οι υπολογιστές να μπορούν να ανταλλάσσουν δεδομένα μεταξύ τους μέσω του Διαδικτύου. Καθορίζεται από το RFC 1122 (Requirements for Internet Hosts – Communication Layers). Τα τέσσερα επίπεδα, από το ψηλότερο στο χαμηλότερο, είναι τα παρακάτω:

* **Εφαρμογής (Application)**Κάθε εφαρμογή έχει προδιαγραφές για την επικοινωνία, έτσι ώστε οι πελάτες και οι εξυπηρετητές να μπορούν να επικοινωνούν υπερπηδώντας διαφορετικές πλατφόρμες. Τα συνήθη πρωτόκολλα περιλαμβάνουν το SSH για απομακρυσμένη σύνδεση, το HTTPS για ασφαλή ιστό, το NFS και το CIFS για διαμοίραση αρχείων και το SMTP για παράδοση ηλεκτρονικού ταχυδρομείου.
* **Μεταφοράς (Transport)**  
  Τα πρωτόκολλα μεταφοράς είναι το TCP και το UDP. Το TCP προσφέρει αξιόπιστη επικοινωνία βασισμένη στη δημιουργία συνδέσεων. Το UDP λειτουργεί στέλνοντας *datagrams* χωρίς τη δημιουργία σύνδεσης. Τα πρωτόκολλα επιπέδου εφαρμογής χρησιμοποιούν θύρες είτε του TCP, είτε του UDP. Υπάρχει μια λίστα γνωστών και καταχωρημένων θυρών που μπορεί να βρεθεί στο αρχείο **/etc/services**.  
    
  Όταν στέλνεται ένα πακέτο στο δίκτυο, ο συνδυασμός της θύρας και της διεύθυνσης IP ονομάζεται *υποδοχή (socket)*. Κάθε πακέτο έχει μια υποδοχή αφετηρίας (source socket) και μια υποδοχή προορισμού. Αυτή η πληροφορία μπορεί να φανεί χρήσιμη στην επιτήρηση και το φιλτράρισμα.
* **Διαδικτύου (Internet)**  
  Το επίπεδο Διαδικτύου (ή δικτύου) είναι υπεύθυνο για τη μεταφορά δεδομένων από τον αρχικό υπολογιστή στον υπολογιστή-προορισμό. Τα πρωτόκολλα του επιπέδου Διαδικτύου είναι τα IPv4 και IPv6. Κάθε υπολογιστής έχει μια διεύθυνση IP και Ενα πρόθεμα που χρησιμεύει στον προσδιορισμό των διευθύνσεων δικτύου. Για τη διασύνδεση μεταξύ δικτύων χρησιμοποιούνται *δρομολογητές (routers)*.
* **Σύνδεσης (Link)**Το επίπεδο σύνδεσης, ή πρόσβασης μέσου, παρέχει τη σύνδεση στο φυσικό μέσο του δικτύου. Οι πιο κοινοί τύποι δικτύου είναι το ενσύρματο Ethernet (802.3) και το ασύρματο τοπικό δίκτυο (Wireless WLAN, 802.11). Κάθε φυσική συσκευή έχει μια διεύθυνση υλικού (hardware address, MAC), η οποία χρησιμοποιείται για να αναγνωριστεί ο προορισμός των πακέτων στο τοπικό τμήμα δικτύου.

### Ονοματοδοσία των θυρών δικτύου

Κάθε δικτυακή διεπαφή (network inteface), δηλαδή κάθε θύρα δικτύου, σε ένα υπολογιστικό σύστημα έχει ένα όνομα, που χρησιμεύει στην ταυτοποίησή της και στον καθορισμό των ρυθμίσεών της.

Σε παλιότερες εκδόσεις του Linux, σαν ονόματα για τις θύρες δικτύου χρησιμοποιούνταν ονόματα όπως **eth0**, **eth1** και **eth2.** Το όνομα **eth0** το έπαιρνε η πρώτη θύρα δικτύου που ανίχνευε το λειτουργικό σύστημα, το **eth1** η δεύτερη, κ.ο.κ.. Το πρόβλημα ήταν ότι με την προσθαφαίρεση συσκευών δικτύου, η διαδικασία ανίχνευσης και ονοματοδοσίας συσκευών μπορούσε να αλλάξει την αντιστοίχηση συσκευής δικτύου και ονόματος. Επιπλέον, το πρότυπο του δίαυλου PCIe, στον οποίο συνδέεται η συντριπτική πλειοψηφία καρτών δικτύου, δεν εγγυάται τη σειρά ανίχνευσης των συσκευών που είναι συνδεμένες στο δίαυλο κατά την εκίνηση του συστήματος. Αυτό μπορεί να προκαλέσει ξαφνικές αλλαγές στην ονοματοδοσία των συσκευών.

Στις νεότερες εκδόσεις του Linux χρησιμοποιείται ένα διαφορετικό σύστημα ονοματοδοσίας, που δε βασίζεται στη σειρά ανίχνευσής τους. Τα ονόματα δίνονται με βάση πληροφορίες από το firmware, την τοπολογία του διαύλου PCI και τον τύπο της συσκευής δικτύου.

Η αρχή των ονομάτων εξαρτάται από τον τύπο της διεπαφής: *en* για ethernet*, wl* για WLAN*, ww* για WWAN.

Το υπόλοιπο του ονόματος εξαρτάται από πληροφορίες που παίρνει το λειτουργικό σύστημα από το firmware του εξυπηρετητή ή καθορίζεται από τη θέση της συσκευής δικτύου στην τοπολογία του PCI.

Το o*N* σημαίνει ότι η συσκευή είναι ενσωματωμένη στη μητρική πλακέτα και ο δείκτης *N* δίνεται από το firmware. eno1 σημαίνει τη συσκευή ethernet 1 που είναι πάνω στη μητρική. Αυτή είναι μια πληροφορία που δε δίνεται πάντα από τους εξυπηρετητές.

Το s*N* σημαίνει ότι η συσκευή είναι στη θέση τύπου hotplug με αριθμό *N* του PCI. ens2 σημαίνει κάρτα ethernet στη θέση hotplug 2του PCI.

Ένα όνομα της μορφής p*M*s*N* περιγράφει μια συσκευή PCI που βρίσκεται στο δίαυλο *M* και στη θέση *N.* Για παράδειγμα, το όνομα wlp1s3 περιγράφει μια κάρτα ασύρματου δικτύου στον τρίτο δίαυλο PCI και στη θέση 3. Για την περίπτωση πολυλειτουργικών καρτών, όπως με πολλαπλές θύρες Ethernet ή με κάρτα Ethernet συν κάποια άλλη λειτουργία (π.χ. Bluetooth), μπορεί να προστεθεί μια κατάληξη της μορφής *fN* στο όνομα: το wwp0s2f1 αναφέρεται στη λειτουργία 1 της κάρτας τύπου WWAN στη θέση 2 του δίαυλου 0. Πιθανότατα αυτή είναι η δεύτερη δικτυακή συσκευή. Η πρώτη κανονικά θα ονομάζεται wwp0s2f0.

Η σταθερή ονοματοδοσία των διεπαφών δικτύου δίνει τη σιγουριά ότι δεν υπάρχει περίπτωση να αλλάξει το όνομα των συσκευών, αλλά επίσης δε θα ονομάζεται απλά eth0 η θύρα ενός συστήματος με μία μόνο θύρα δικτύου.

### Δίκτυα IPv4

Το IPv4 αποτελεί το κυριότερο πρωτόκολλο που χρησιμοποιείται σήμερα στο Διαδίκτυο. Για να μπορεί κανείς να διαχειριστεί τη δικτυακή επικοινωνία των εξυπηρετητών, χρειάζεται να έχει τουλάχιστον βασική κατανόηση του τρόπου οργάνωσης και λειτουργίας των δικτύων IPv4.

#### Οι διευθύνσεις στο IPv4

Οι διευθύνσεις στα δίκτυα IPv4 είναι αριθμοί των 32 bit, που τους γράφουμε σαν τέσσερεις δεκαδικές οκτάδες (άρα κάθε οκτάδα παίρνει τιμές μεταξύ 0 και 255). Οι οκτάδες χωρίζονται μεταξύ τους με τελείες. Η διεύθυνση χωρίζεται σε δύο λογικά μέρη: το *μέρος δικτύου – network part* και το *μέρος υπολογιστή – host part.* Όλοι οι υπολογιστές που έχουν το ίδιο μέρος δικτύου ανήκουν στο ίδιο υποδίκτυο και μπορούν να επικοινωνούν μεταξύ τους άμεσα, χωρίς δρομολογητή (router). Το μέρος υπολογιστή, είναι μοναδικό για κάθε υπολογιστή σε ένα υποδίκτυο και χρησιμοποιείται για την αναγνώριση κάθε υπολογιστή του υποδικτύου.

Από όλες τις διευθύνσεις ενός υποδικτύου, δύο έχουν ειδική σημασία. Η μικρότερη δυνατή διεύθυνση, όπου όλα τα bit υπολογιστή είναι μηδέν ονομάζεται συνήθως *διεύθυνση δικτύου*. Η μεγαλύτερη δυνατή, αυτή με όλα τα bit υπολογιστή να είναι ένα, χρησιμοποιείται για μετάδοση μηνυμάτων broadcast (προς όλα τα μέλη του δικτύου) και δεν μπορεί να αποδοθεί σε κανέναν υπολογιστή.

Σε αντίθεση με παλιότερα, στα σύγχρονα δίκτυα (και το Διαδίκτυο) το μέγεθος ενός υποδικτύου IPv4 είναι μεταβλητό. Για να ξέρουμε ποιό μέρος μιας διεύθυνσης IPv4 είναι αυτό του δικτύου και ποιό του υπολογιστή, χρειαζόμαστε την επιπλέον πληροφορία της *μάσκας δικτύου – netmask* που έχει το υποδίκτυο. Αυτή μας δείχνει πόσα bits της διεύθυνσης ανήκουν στο υποδίκτυο. Τα υπόλοιπα bits ανήκουν στο μέρος υπολογιστή και ο αριθμός τους καθορίζει μέχρι πόσοι υπολογιστές μπορούν να συνδεθούν στο υποδίκτυο.

Οι μάσκες δικτύου παλιότερα γράφονταν με μορφή παρόμοια με τις διευθύνσεις IPv4, σαν τέσσερεις δεκαδικές οκτάδες που περιγράφουν το τμήμα δικτύου. Για παράδειγμα, η μάσκα δικτύου των 24bit γράφεται σαν 255.255.255.0. Ο νεότερος τρόπος σύνταξης για την ίδια μάσκα ορίζει ένα *πρόθεμα δικτύου* /24. Αυτός ο τρόπος σύνταξης ονομάζεται CIDR. Οι δύο τρόποι είναι ισοδύναμοι.

Παρακάτω δίνονται δύο παραδείγματα ορισμού δικτύων.

|  |  |
| --- | --- |
| **Διεύθυνση IPv4** | 10.63.14.77 |
| **Δυαδική μορφή** | 00001010.00111111.00001110.01001110 |
| **Μάσκα δικτύου** | 255.255.0.0 |
| **Πρόθεμα** | /16 |
| **Δυαδική μορφή δικτύου** | 00001010.00111111 |
| **Δυαδική μορφή υπολογιστή** | 00001110.01001110 |

Πίνακας . Δίκτυο 10.63.0.0/16

|  |  |
| --- | --- |
| **Διεύθυνση IPv4** | 192.168.1.7 |
| **Δυαδική μορφή** | 11000000.10101000.00000001.00000111 |
| **Μάσκα δικτύου** | 255.255.255.0 |
| **Πρόθεμα** | /24 |
| **Δυαδική μορφή δικτύου** | 11000000.10101000 |
| **Δυαδική μορφή υπολογιστή** | 00000001.00000111 |

Πίνακας . Δίκτυο 192.168.0.0/24

Στο IPv4 έχουν καθοριστεί τρεις ζώνες διευθύνσεων οι οποίες προορίζονται για χρήση μόνο σε ιδιωτικά δίκτυα. Αυτές ονομάζονται *ιδιωτικές* διευθύνσεις (*private* addresses), ενώ όλες οι υπόλοιπες ονομάζονται *δημόσιες* (*public*). Η χρήση τους δεν επιτρέπεται στο δημόσιο Διαδίκτυο και δεν γίνεται δρομολόγηση γι΄ αυτές.

Εκτός των άλλων, έτσι απομονώνονται οι υπολογιστές και άλλες δικτυωμένες συσκευές ενός ή εταιρείας από το Διαδίκτυο. Για να μπορούν αυτές να έχουν πρόσβαση στο διαδίκτυο, αλλά και να υπάρχει πρόσβαση σε όποιες από αυτές είναι επιθυμητό από το Διαδίκτυο, εφαρμόζονται τεχνικές όπως η **μετάφραση διευθύνσεων δικτύου** (network address translation, NAT), όπου εκεί που συναντιούνται τα δύο δίκτυα (ιδιωτικό και Διαδίκτυο) ιδιωτικές διευθύνσεις αντιστοιχίζονται σε δημόσιες διευθύνσεις και, ανάλογα με την κατεύθυνση της κίνησης, η μία διεύθυνση αλλάζει με την αντίστοιχή της.

Οι τρεις ζώνες *ιδιωτικών* διευθύνσεων φαίνονται στον παρακάτω πίνακα:

|  |  |  |  |
| --- | --- | --- | --- |
| Ζώνη διευθύνσεων | Μάσκα | Εύρος διευθύνσεων | Αριθμός διευθύνσεων |
| 10.0.0.0/8 | 255.0.0.0 | 1.0.0.0–10.255.255.255 | 16777216 |
| 172.16.0.0/12 | 255.240.0.0 | 172.16.00–172.31.255.255 | 1048576 |
| 192.168.0.0/24 | 255.255.255.0 | 192.168.0.0–192.168.255.255 | 65536 |

Πίνακας 4.3 Ζώνες ιδιωτικών διευθύνσεων του IPv4

Στους επόμενους πίνακες φαίνονται παραδείγματα υπολογισμού της διεύθυνσης δικτύου για αυτές τις τρεις ζώνες. Σημειώστε ότι οι μάσκες που αναφέρονται στον προηγούμενο πίνακα είναι οι μικρότερες δυνατές για την κάθε ζώνη και μπορούν να προσαρμόζονται ανάλογα με τις απαιτήσεις.

|  |  |  |
| --- | --- | --- |
| Διεύθυνση υπολογιστή | 192.168.127.65 | 11000000.10101000.01111111.01000001 |
| Πρόθεμα δικτύου | /24 (255.255.255.0) | 11111111.11111111.11111111.00000000 |
| Διεύθυνση δικτύου | 192.168.127.0 | 11000000.10101000.01111111.00000000 |
| Διεύθυνση εκπομπής (broadcast) |  |  |

|  |  |  |
| --- | --- | --- |
| Διεύθυνση υπολογιστή | 172.17.18.1 | 10101100.00010001.00010010.00000001 |
| Πρόθεμα δικτύου | /20 (255.255.240.0) | 11111111.11111111.11110000.00000000 |
| Διεύθυνση δικτύου | 172.17.16.0 | 10101100.00010001.00010000.00000000 |
| Διεύθυνση εκπομπής (broadcast) | 172.17.31.255 | 10101100.00010001.00011111.11111111 |

|  |  |  |
| --- | --- | --- |
| Διεύθυνση υπολογιστή | 10.1.2.3 | 00001010.00000001.00000010.00000011 |
| Πρόθεμα δικτύου | /8 (255.0.0.0) | 11111111.00000000.00000000.00000000 |
| Διεύθυνση δικτύου | 10.0.0.0 | 00001010.00000000.00000000.00000000 |
| Διεύθυνση εκπομπής (broadcast) | 10.255.255.255 | 00001010.11111111.11111111.11111111 |

Τέλος, υπάρχει μια ειδική διεύθυνση, η 127.0.0.1, η οποία πάντα δείχνει στο τοπικό σύστημα (συνήθως αναφέρεται και σαν localhost), στο οποίο ανήκει και όλο το δίκτυο 127.0.0.0/8. Αυτό δίνει τη δυνατότητα στο τοπικό σύστημα τόσο να μπορεί να συνομιλεί με τον εαυτό του μέσω πρωτοκόλλων δικτύου, όσο και να μπορεί να επιβεβαιώσει ότι το δίκτυο στο τοπικό σύστημα λειτουργεί στοιχειωδώς σωστά (όσον αφορά το λειτουργικό σύστημα και τις βασικές του ρυθμίσεις) χωρίς να είναι απαραίτητη η ύπαρξη άλλων συστημάτων ή συσκευών δικτύου.

#### Δρομολόγηση στο IPv4

Η δρομολόγηση εξασφαλίζει τη μετακίνηση της δικτυακής κυκλοφορίας από υπολογιστή σε υπολογιστή και από ένα δίκτυο σε ένα άλλο.

Ξεκινώντας από έναν υπολογιστή, σε αυτόν υπάρχει ένας *πίνακας δρομολόγησης – routing table*, που περιγράφει το πώς θα δρομολογηθεί η κίνηση προς συγκεκριμένα δίκτυα. Μια εγγραφή στον πίνακα αναφέρει ένα δίκτυο προορισμού, ποια διεπαφή θα χρησιμοποιηθεί για την αποστολή της κίνησης, καθώς και τη διεύθυνση IP του ενδιάμεσου δρομολογητή, που τυχαίνει να χρειάζεται για να μεταβιβάσει ένα μήνυμα προς τον προορισμό του. Χρησιμοποιείται η εγγραφή του πίνακα δρομολόγησης που ταιριάζει στον προορισμό του μηνύματος, ενώ, σε περίπτωση που ταιριάζουν περισσότερες εγγραφές, χρησιμοποιείται αυτή που έχει το μεγαλύτερο μήκος προθέματος, σαν πιο ειδική (ταιριάζει καλύτερα).

Στον πίνακα δρομολόγησης συνήθως υπάρχει και μια εγγραφή για μια *προεπιλεγμένη διαδρομή – default route*, που έχει σαν δίκτυο προορισμού ολόκληρο το Διαδίκτυο του IPv4, συγκεκριμένα το δίκτυο 0.0.0.0/0, και δείχνει σε έναν δρομολογητή προσπελάσιμο από κάποιο υποδίκτυο και ο οποίος θα έχει μια πιο ειδική διαδρομή προς τον συγκεκριμένο προορισμό.

Οι υπολογιστές, όταν λαβαίνουν κάποια πακέτα που δεν απευθύνονται σε αυτούς, συνήθως τα απορρίπτουν. Αντίθετα, οι δρομολογητές τα *προωθούν* με βάση το δικό τους πίνακα δρομολόγησης. Αυτό μπορεί να σημαίνει την αποστολή του πακέτου στον υπολογιστή προορισμού του, αν τυχαίνει να είναι σε υποδίκτυο που ανήκει και ο δρομολογητής, ή την περαιτέρω προώθησή του σε κάποιον άλλο δρομολογητή. Η διαδικασία επαναλαμβάνεται μέχρι το πακέτο να φτάσει στον προορισμό του.

A computer screen shot of a computer

AI-generated content may be incorrect.

Εικόνα 1 - Παράδειγμα τοπολογίας δικτύου

**παράδειγμα πίνακα δρομολόγησης**

|  |  |  |
| --- | --- | --- |
| ΠΡΟΟΡΙΣΜΟΣ | ΔΙΕΠΑΦΗ | ΔΡΟΜΟΛΟΓΗΤΗΣ |
| 192.168.0.0/24 | wlp0 | --- |
| 192.168.1.0/24 | enp0s25 | --- |
| 0.0.0.0/0 (προεπιλεγμένος) | enp0s25 | 192.168.1.1 |

#### Ρύθμιση διευθύνσεων IPv4 και δρομολόγησης

Ένας εξυπηρετητής μπορεί να θέσει αυτόματα τις ρυθμίσεις δικτύου IPv4 κατά την εκκίνηση παίρνοντας πληροφορίες από έναν *εξυπηρετητή DHCP*, αν είναι ρυθμισμένο το DHCP και τρέχει ο αντίστοιχος daemon τοπικά. Αυτός εκπέμπει ένα ερώτημα από τη σύνδεση δικτύου για τοπικό εξυπηρετητή και ρυθμίσεις δικτύου (διεύθυνση IPv4, μάσκα, κλπ.) και λαβαίνει μια  *μίσθωση – lease* για συγκεκριμένο χρόνο για τον οποίο μπορεί να χρησιμοποιήσει τις συγκεκριμένες ρυθμίσεις. Η μίσθωση πρέπει να ανανεώνεται περιοδικά, αλλιώς οι ρυθμίσεις χάνονται και μπορούν να δοθούν σε άλλο μηχάνημα.

### Δίκτυα IPv6

Το IPv6 προορίζεται να είναι ο αντικαταστάτης του IPv4. Μια και υπάρχουν συστήματα που χρησιμοποιούν ήδη το IPv6 και τα οποία αυξάνονται με τον καιρό, μια βασική κατανόηση της λειτουργίας του είναι χρήσιμη. Πολλοί οργανισμοί το χρησιμοποιούν ήδη εσωτερικά στα δίκτυά τους Π.χ. πάροχοι υπηρεσιών Διαδικτύου το χρησιμοποιούν, ώστε να μην καταναλώνουν πολύτιμες (λόγω έλλειψης) διευθύνσεις IPv4.

Το IPv6 μπορεί και να χρησιμοποιείται παράλληλα με το IPv4, σε αυτό που ονομάζεται μοντέλο *dual stack*. Σε αυτόν τον τρόπο λειτουργίας, οι διασυνδέσεις δικτύου μπορούν να έχουν ταυτόχρονα διεύθυνση ή διευθύνσεις IPv6 και διευθύνσεις IPv4. Πολλοί εξυπηρετητές Linux λειτουργούν έτσι από προεπιλογή, όπως π.χ. το Rocky Linux.

#### Διευθύνσεις IPv6

Οι διευθύνσεις IPv6 είναι αριθμοί μήκους 128 bit. Η συνηθισμένη μορφή απεικόνισής τους είναι αυτή τετράδων δεκαεξαδικών nibbles (nibble: μισό byte, 4 bit) χωρισμένες μεταξύ τους με άνω-κάτω τελείες. Κάθε τετράδα αντιστοιχεί λοιπόν σε 16 bit.

Για λόγους ευκολίας στο γράψιμο των διευθύνσεων, ακολουθούνται οι εξής κανόνες:

* Αν στις τετράδες τα πιο σημαντικά ψηφία είναι μηδέν, τότε παραλείπονται, αλλά με τον περιορισμό να γράφεται τουλάχιστο ένα ψηφίο ανά τετράδα. Αν λοιπόν μια τετράδα είναι μηδενική, τότε γράφεται τουλάχιστο ένα μηδενικό, π.χ.: a1b3:de:0:f3a:0:0:0:96.
* Αν σε μια διεύθυνση υπάρχουν μακριές ακολουθίες από μηδενικά, τότε μία ή περισσότερες διαδοχικές τετράδες μηδενικών συνδυάζονται σε *ένα μόνο* μπλοκ ::. Αν υπάρχουν περισσότερες από μία τέτοιες ομάδες, προτιμάμε να το κάνουμε στη μεγαλύτερη. Αν είναι ισομεγέθεις, προτιμάμε να το εφαρμόζουμε στην ομάδα που είναι πιο αριστερά (πιο σημαντικά ψηφία). Επίσης, προτιμάμε να μην το εφαρμόζουμε σε μονές τετράδες μηδενικών, αλλά σε δύο ή περισσότερες συνεχόμενες. Η διεύθυνση που γράψαμε παραπάνω γίνεται a1b3:de:0:f3a::96, αλλά και η γραφή a1b3:de::f3a:0:0:0:96 είναι επίσης έγκυρη, αλλά λιγότερο βολική.
* Χρησιμοποιούμε μικρά γράμματα για τα γράμματα στους δεκαεξαδικούς αριθμούς.
* Αν μαζί με τη διεύθυνση περιλαμβάνεται και κάποια θύρα δικτύου TCP ή UDP, για να ξεχωρίζει, περικλείουμε τη διεύθυνση σε αγκύλες: [a1b3:de:0:f3a::96]:443.

#### Υποδίκτυα στο IPv6

Σε κάποια αντιστοίχιση με το IPv4, μια διεύθυνση IPv6 χωρίζεται σε δύο μέρη: το *πρόθεμα δικτύου – network prefix* και το *αναγνωριστικό διεπαφής – interface ID*. Το πρόθεμα δικτύου ταυτοποιεί το υποδίκτυο και δεν επιτρέπεται σε διεπαφές δικτύου που ανήκουν στο ίδιο υποδίκτυο να έχουν το ίδιο αναγνωριστικό διεπαφής, μια και το αναγνωριστικό χρησιμοποιείται για να ταυτοποιεί μία συγκεκριμένη διεπαφή στο υποδίκτυο. Η διεύθυνση αυτή είναι του τύπου που ονομάζεται *global unicast*, δηλαδή αποτελεί μοναδική διεύθυνση της διεπαφής για ολόκληρο το Διαδίκτυο.

Εκεί που διαφέρει το IPv6 από το IPv4 είναι στο ότι έχει μια σταθερή μάσκα δικτύου, την /64, η οποία χρησιμοποιείται σχεδόν σε όλες τις κανονικές διευθύνσεις IPv6. Έτσι, το πρόθεμα δικτύου και το αναγνωριστικό διεπαφής μοιράζονται εξίσου τη διεύθυνση. Ο συνολικός αριθμός 232 διεπαφών ανά υποδίκτυο δίνει τεράστιο χώρο για το συνολικό αριθμό υπολογιστών σε ένα υποδίκτυο. Αν υπολογίσουμε ότι και τα δυνατά υποδίκτυα είναι επίσης 232,, μπορούμε να υποθέσουμε ότι δε θα ξεμείνουμε εύκολα από διευθύνσεις δικτύου, όπως έγινε με το IPv4.

Συνήθως οι πάροχοι δικτύων κατανέμουν ένα μικρότερο πρόθεμα σε κάποιον οργανισμό, π.χ. /48. Σε αυτή την περίπτωση ο οργανισμός έχει τα υπόλοιπα 16 bit του προθέματος δικτύου για να φτιάξει έως και 65536 υποδίκτυα.

Το αντίστοιχο του 127.0.0.1/8 του IPv4 στο IPv6 είναι το ::1/128, ενώ το 0.0.0.0 στο IPv6 γίνεται ::. Αυτό το τελευταίο χρησιμοποιείται τόσο σε δικτυακές υπηρεσίες, ώστε να δείχνουν ότι ακούν σε όλες τις ρυθμισμένες διευθύνεις IP, όσο και για την προεπιλεγμένη διαδρομή σε πίνακες δρομολόγησης (με τη μορφή ::/0).

Οι διευθύνσεις τύπου global unicast που αποδίδει το IANA ανήκουν στο διάστημα 2000::/3 και ισοδυναμούν με τα δίκτυα 2000::/16 έως και 3fff::/16.

Στο IPv6 υπάρχει ένα είδος διευθύνσεων που ονομάζονται *μοναδικές τοπικές διευθύνσεις – unique local addresses*. Είναι οι διευθύνσεις fd00::/8 και χρησιμοποιούνται στο IPv6 έτσι ώστε να μπορεί ένας οργανισμός να κατανείμει ένα ιδιωτικό διάστημα διευθύνσεων σε δίκτυα που να μπορούν να δρομολογηθούν εντός του οργανισμού, αλλά να μην μπορούν να χρησιμοποιηθούν στο παγκόσμιο Διαδίκτυο. Αυτό γίνεται επιλέγοντας *τυχαία* από αυτό το διάστημα ένα τμήμα /48, το οποίο μπορεί να χωριστεί κανονικά σε δίκτυα /64. Στο IPv6 δεν υπάρχει ακριβές αντίστοιχο των ιδιωτικών διαστημάτων διευθύνσεων του IPv4 (192.168.0.0/16, 172.16.0.0/12 και 10.0.0.0/8).

Οι διευθύνσεις τύπου *τοπικής σύνδεσης* (*link-local)* του IPv6 ανήκουν στο διάστημα fe80::/10 και αποδίδονται αυτόματα σε κάθε διεπαφή IPv6. Λειτουργούν μόνο στην τοπική *σύνδεση – link* (π.χ. ένα VLAN του Ethernet) στο δίκτυο fe80::/64 και δεν μπορούν να δρομολογηθούν έξω από αυτή. Μπορούν να χρησιμοποιηθούν από τους άλλους υπολογιστές που ανήκουν στην ίδια σύνδεση. Στην απόδοση διεύθυνσης τοπικής σύνδεσης (link-local) χρησιμοποιείται η διεύθυνση MAC του Ethernet. Για να γίνουν τα 48 bit της διεύθυνσης MAC 64 για το αναγνωριστικό διεπαφής, συνήθως αντιστρέφεται το bit 7 της διεύθυνσης MAC και εισάγεται η τιμή ff:fe ανάμεσα στα δύο μεσαία bytes της. Για παράδειγμα, η διεύθυνση MAC 88:77:66:cc:dd:ee με τη συγκεκριμένη διαδικασία μετατρέπεται στη διεύθυνση link-local fe80::8a77:66ff:fecc:ddee/64. Επειδή οι διευθύνσεις link-local δεν δρομολογούνται, για να επικοινωνήσει ένας υπολογιστής με μια διεύθυνση link-local πρέπει να προσδιορίσει τη σύνδεση (link) που πρέπει να χρησιμοποιηθεί. Αυτό γίνεται με ένα *αναγνωριστικό εμβέλειας – scope identifier*, που προστίθεται στο τέλος της διεύθυνσης και αποτελείται από το % και τη διεπαφή δικτύου. Έτσι, για να κάνουμε ping στη διεύθυνση linl-local του παραδείγματος, μέσω του συνδέσμου που είναι συνδεμένος στη διεπαφή δικτύου eno1, χρησιμοποιώντας την εντολή ping6, γράφουμε τα παρακάτω στη γραμμή εντολής:

**user@linuxsrv ~ $** ping6 fe80::8a77:66ff:fecc:ddee%enp0s25

Το δίκτυο Multicast στο IPv6 (αντίστοιχο του 224.0.0.0/4 του IPv4) είναι το ff00::/8 και χρησιμεύει στο να εκπέμπουμε πακέτα σε πολλούς υπολογιστές την ίδια στιγμή. Σε αντίθεση με την *εκπομπή – broadcast*  του IPv4 (που δεν έχει αντίστοιχο στο IPv6), τα πακέτα κατευθύνονται μόνο σε συγκεκριμένα μηχανήματα, ενώ κάποια πακέτα multicast μπορούν να δρομολογηθούν σε άλλα υποδίκτυα, αναλόγως των ρυθμίσεων των δρομολογητών και των συστημάτων. Για να σταλεί multicast σε μια σύνδεση (link) για διευθύνεις τύπου link-local, χρησιμοποιείται η διεύθυνση multicast ff02::1, που είναι η διεύθυνση link-local all-nodes (προς όλους τους κόμβους). Αντίστοιχα με το παράδειγμα για τις διευθύνσεις link-local, κάνουμε ping χρησιμοποιώντας τη διεύθυνση multicast για link-local προσθέτοντας το % και τη δικτυακή διεπαφή enp0s25 στο τέλος της διεύθυνσης:

**user@linuxsrv ~ $** ping6 ff02::1%enp0s25

PING ff02::1%enp0s25 (ff02::1%enp0s25) 56 data bytes

64 bytes from fe80::f2de:f1ff:fe5f:227%enp0s25: icmp\_seq=1 ttl=64 time=0.056 ms

64 bytes from fe80::2517:99a1:92fc:bd49%enp0s25: icmp\_seq=1 ttl=64 time=0.630 ms

64 bytes from fe80::7ac5:7dff:fef4:3e30%enp0s25: icmp\_seq=1 ttl=64 time=1.26 ms

64 bytes from fe80::4f4e:4a4c:4db3:2e2b%enp0s25: icmp\_seq=1 ttl=64 time=3.74 ms

64 bytes from fe80::ee88:92ff:fed2:b1db%enp0s25: icmp\_seq=1 ttl=64 time=104 ms

( … )

#### Ρύθμιση διευθύνσεων IPv6

Στο IPv6, μια διεύθυνση μπορεί να αποδοθεί χειροκίνητα από τον διαχειριστή και αυτόματα με δύο τρόπους: το DHCPv6 και το SLAAC.

Στη χειροκίνητη ανάθεση διευθύνσεων υπάρχουν κάποια αναγνωριστικά διεπαφής που δεν μπορούν να χρησιμοποιηθούν:

* Το αναγνωριστικό με όλα τα ψηφία μηδενικά 0000:0000:0000 ονομάζεται subnet router anycast και χρησιμοποιείται από όλους τους δρομολογητές στη σύνδεση.
* Τα αναγνωριστικά fdff:ffff:ffff:ff80 έως και fdff:ffff:ffff:ffff.

Επειδή στο IPv6 δεν υπάρχει διεύθυνση εκπομπής – broadcast, τα αιτήματα του DHCPv6 στέλνονται στη θύρα UDP 547 του ff02::1:2, που είναι η ομάδα multicast all-dhcp-servers για την τοπική σύνδεση (link-local), άρα ακούνε σε αυτήν όλοι οι εξυπηρετητές DHCP της τοπικής σύνδεσης. Ο εξυπηρετητής DHCPv6 απαντάει τότε στέλνοντας τις κατάλληλες πληροφορίες στη θύρα 546 UDP της διεύθυνσης link-local του μηχανήματος που έστειλε το αίτημα.

(Ο εξυπηρετητής για το DHCPv6 περιλαμβάνεται στο πακέτο *dhcp*)

Με το SLAAC (Stateless Address Autoconfiguration), ο υπολογιστής ενεργοποιεί τη διεπαφή του με μια διεύθυνση link-local και μετά στέλνει ένα μήνυμα «router solicitation» στη διεύθυνση ff02::2, την τοπική ομάδα multicast για link-local όλων των δρομολογητών (all-routers link-local multicast group). Κάποιος δρομολογητής IPv6 στην τοπική σύνδεση στέλνει απάντηση στη διεύθυνση link-local του υπολογιστή με ένα πρόθεμα δικτύου και ίσως κάποιες επιπλέον πληροφορίες. Ο υπολογιστής χρησιμοποιεί αυτό το πρόθεμα δικτύου μαζί με ένα αναγνωριστικό διεπαφής που φτιάχνει, συνήθως ακολουθώντας τα ίδια βήματα που χρησιμοποιεί για τη δημιουργία διευθύνσεων linl-local. Ο δρομολογητής στέλνει περιοδικά ενημερώσεις multicast, τις ονομαζόμενες «router advertisements», για επιβεβαίωση ή ενημέρωση των πληροφοριών που παρείχε.

(Ο εξυπηρετητής για το SLAAC παρέχεται από το πακέτο *radvd*)

Χρειάζεται προσοχή ως προς το ποια από τα DHCP, DHCPv6 και SLAAC είναι ενεργοποιημένα σε έναν υπολογιστή ή εξυπηρετητή, ώστε να μην αποκτήσει π.χ. διεύθυνση IPv6 χωρίς να το θέλουμε.

### Ονόματα υπολογιστών και διευθύνσεις IP

Γενικά οι άνθρωποι προτιμάμε να χρησιμοποιούμε ονόματα και όχι αριθμούς. Για το λόγο αυτό, υπάρχουν στο Linux (και όχι μόνο! ) μια σειρά μηχανισμοί που αντιστοιχίζουν ένα όνομα υπολογιστή με μια διεύθυνση IP. Η ονομασία τους είναι *name resolution (επίλυση ονόματος).*

Ο πρώτος τρόπος για να βρούμε την αντιστοίχιση ονομάτων – διευθύνσεων IP είναι με την εισαγωγή εγγραφών στο αρχείο **/etc/hosts** κάθε συστήματος. Αυτός ο τρόπος σημαίνει ότι πρέπει να ανανεώνεται χειροκίνητα αυτό το αρχείο σε κάθε εξυπηρετητή. Εκτός από χρονοβόρα σαν μέθοδος, είναι και επιρρεπής σε λάθη.

Τις περισσότερες περιπτώσεις, μπορεί να βρει κανείς τη διεύθυνση που αντιστοιχεί σε ένα όνομα μέσω μιας δικτυακής υπηρεσίας που ονομάζεται *Domain Name System* (DNS). Αποτελείται από ένα κατανεμημένο δίκτυο εξυπηρετητών που παρέχουν τις αντιστοιχίσεις ονομάτων υπολογιστών σε διευθύνσεις IP. Για έναν υπολογιστή που χρειάζεται να βρει μια τέτοια αντιστοίχιση, αρκεί να γνωρίζει έναν τέτοιο εξυπηρετητή, ένα *nameserver*.Ο εξυπηρετητής δε χρειάζεται να είναι στο ίδιο υποδίκτυο με τον υπολογιστή, μόνο να είναι προσβάσιμος από αυτόν. Αυτό γίνεται συνήθως είτε μέσω του DHCP, είτε από μια στατική ρύθμιση στο αρχείο **/etc/resolv.conf**. Η ρύθμιση του name resolution αναλύεται σε επόμενη ενότητα του κεφαλαίου.

### Ερωτήσεις ενότητας

1. Από πόσα bit αποτελείται μια διεύθυνση IPv4;
   1. 8
   2. 64
   3. **32**
   4. 16
   5. 4
   6. 128
2. Ποιος όρος καθορίζει πόσα από τα πιο σημαντικά bit της διεύθυνσης IP αποτελούν τη διεύθυνση δικτύου της;
   1. εμβέλεια δικτύου
   2. **μάσκα δικτύου**
   3. υποδίκτυο
   4. δίκτυο
   5. multicast
   6. διεύθυνση δικτύου
3. Ποια από τις παρακάτω είναι έγκυρη διεύθυνση IPv4 για υπολογιστή;
   1. 192.168.0.0
   2. **192.168.0.65**
   3. 192.168.0.256
   4. 192.168.0.255
4. Από πόσα bit αποτελείται μια διεύθυνση IPv6;
   1. 8
   2. 64
   3. 32
   4. 16
   5. 4
   6. **128**
5. Ποια από τις παρακάτω διευθύνσεις δεν είναι έγκυρη διεύθυνση IPv6;
   1. 8::4
   2. 2025:bad::c1:4
   3. ff02:a::0:0
   4. 2002::8
   5. **2025:777::ace::1**
   6. 2025:0000:0000:0000:0000:0000:0000:1000
6. Ποιος όρος επιτρέπει σε έναν υπολογιστή να στέλνει πακέτα σε μια ειδική διεύθυνση IP και τα οποία παραλαμβάνονται από πολλαπλούς υπολογιστές;
   1. εμβέλεια δικτύου
   2. μάσκα δικτύου
   3. υποδίκτυο
   4. δίκτυο
   5. **multicast**
   6. διεύθυνση δικτύου

## Επιβεβαιώνοντας τις ρυθμίσεις δικτύου

Σε αυτή την ενότητα θα μιλήσουμε για τον έλεγχο και την επιθεώρηση των τρεχουσών ρυθμίσεων δικτύου με χρήση της γραμμής εντολών.

### Συλλέγοντας πληροφορίες για τις διεπαφές δικτύου

Παλιότερα οι πληροφορίες που χρειαζόμαστε για το δίκτυο μπορούσαν να βρεθούν με εντολές όπως η **ifconfig**, η **route**, κλπ.. Οι εντολές αυτές εξακολουθούν να ισχύουν, αλλά προτιμάται ο νέος τρόπος, αυτός της εντολής **ip**. Είναι μια εντολή-ομπρέλα, που μέσω των παραμέτρων της αντικαθιστά μια ομάδα εντολών. Η σύνταξη των παλαιότερων εντολών μοιάζει πολύ με αυτή των νέων, τόσο ώστε να δείχνουν και οι μεν και οι δε οικείες και κατανοητές, αλλά ταυτόχρονα είναι αρκετά διαφορετικές, ώστε να είναι εκνευριστικό όταν εφαρμόζει κανείς κατά λάθος τη σύνταξη της μιας στην άλλη.

#### Αναγνώριση των διεπαφών δικτύου

Η εντολή **ip link** δίνει μια λίστα με όλες τις διεπαφές δικτύου που υπάρχουν στο σύστημα:

**user@linuxsrv ~ $** ip link show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group default qlen 1000

link/ether 08:11:96:01:39:88 brd ff:ff:ff:ff:ff:ff

Βλέπουμε ότι υπάρχουν τρεις διεπαφές: η lo, που είναι η συσκευή loopback και συνδέεται με τον ίδιο τον υπολογιστή, μια διεπαφή ethernet (enp0s25) και μια Wi-Fi (wlp3s0).

Για να ρυθμιστούν σωστά οι διεπαφές, πρέπει να γνωρίζουμε σε ποιο δίκτυο είναι συνδεμένη η κάθε μία τους. Συνήθως είναι γνωστή και η διεύθυνση MAC της διεπαφής, είτε επειδή αναγράφεται στο ίδιο το μηχάνημα ή την κάρτα δικτύου, είτε επειδή είναι εικονικό μηχάνημα και γνωρίζουμε τις ρυθμίσεις του. Στο παραπάνω παράδειγμα η διεύθυνση MAC φαίνεται αμέσως δεξιά της ένδειξης «link/ether».

#### Εύρεση των διευθύνσεων IP

Πάλι με τη εντολή ip μπορούμε να δούμε πληροφορίες για τις συσκευές και τη διεύθυνσή τους.

**user@linuxsrv ~ $** ip addr show enp0s25

3: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.71/24 brd 192.168.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet6 2a02:2149:8a6a:df00:f2de:f1ff:fe5f:227/64 scope global dynamic mngtmpaddr proto kernel\_ra

valid\_lft 4294955147sec preferred\_lft 4294955147sec

inet6 fe80::f2de:f1ff:fe5f:227/64 scope link proto kernel\_ll

valid\_lft forever preferred\_lft forever

Ας εξετάσουμε τα τμήματα που έχουν επισημανθεί στο παραπάνω παράδειγμα.

Βλέπουμε στην πρώτη γραμμή τη λέξη UP, που μας δείχνει ότι η διεπαφή είναι ενεργή, σε λειτουργία.

Στην τρίτη γραμμή (link/ether) βλέπουμε τη διεύθυνση MAC της συσκευής.

Στην αμέσως επόμενη (inet) βλέπουμε τη διεύθυνση IPv4, το μήκος του προθέματος δικτύου και την εμβέλειά της («global»).

Δύο γραμμές παρακάτω βλέπουμε την διεύθυνση IPv6 της διεπαφής, το μήκος του προθέματος δικτύου και την εμβέλεια, που είναι και εδώ global, άρα έχει κανονική χρήση.

Η τελευταία διεύθυνση που βλέπουμε είναι και αυτή IPv6, αλλά η εμβέλειά της είναι «link», που σημαίνει ότι μπορεί να χρησιμοποιηθεί για επικοινωνία μόνο εντός του τοπικού συνδέσμου Ethernet (βλ. και την παραπάνω ενότητα για το IPv6).

Να επισημάνουμε εδώ ότι όπως η διεπαφή έχει περισσότερες από μία διευθύνεις IPv6, έτσι μπορεί να έχει και περισσότερες από μία διευθύνσεις IPv4. Απλώς υπάρχουν κάποιοι περιορισμοί, π.χ. μπορεί να υπάρχει μόνο μία *default gateway – προεπιλεγμένη πύλη εξόδου*. Η δρομολόγηση για τα διαφορετικά δίκτυα που πιθανό να ανήκει κάθε διεύθυνση μπορεί να γίνει μέσω εγγραφών στον πίνακα δρομολόγησης.

#### Εμφανίζοντας στατιστικά απόδοσης

Με μια επιλογή στη γραμμή εντολής, η εντολή **ip** μας εμφανίζει στατιστικά στοιχεία, όπως διάφορους μετρητές για κάθε διεπαφή, όπως για αριθμό ληφθέντων και απεσταλμένων πακέτων, λαθών στα πακέτα, πακέτα που απορρίφθηκαν, κλπ.. Οι ενδείξεις τους μπορούν να χρησιμοποιηθούν για την ανίχνευση προβλημάτων στο δίκτυο.

**user@linuxsrv ~ $** ip -s link show enp0s25

3: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

RX: bytes packets errors dropped missed mcast

2726761940 1987909 0 0 0 3252

TX: bytes packets errors dropped carrier collsns

136747042 1053579 0 9 0 0

### Έλεγχος συνδεσιμότητας μεταξύ υπολογιστών

Για τον έλεγχο της συνδεσιμότητας χρησιμοποιείται η εντολή **ping**. Η εντολή συνεχίζει να τρέχει μέχρι να πατήσουμε **Ctrl+C**, εκτός κι αν έχουν περαστεί επιλογές στη γραμμή εντολής για τον περιορισμό των πακέτων που θα αποσταλούν:

**user@linuxsrv ~ $** ping 192.168.1.65

PING 192.168.1.65 (192.168.1.65) 56(84) bytes of data.

64 bytes from 192.168.1.65: icmp\_seq=1 ttl=64 time=0.201 ms

64 bytes from 192.168.1.65: icmp\_seq=2 ttl=64 time=0.278 ms

64 bytes from 192.168.1.65: icmp\_seq=3 ttl=64 time=0.286 ms

64 bytes from 192.168.1.65: icmp\_seq=4 ttl=64 time=0.266 ms

^C

--- 192.168.1.65 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3089ms

rtt min/avg/max/mdev = 0.201/0.257/0.286/0.033 ms

Η αντίστοιχη εντολή για το IPv6 είναι η εντολή **ping6**:

**user@linuxsrv ~ $** ping6 fe80::2517:99a1:92fc:bd49

PING fe80::2517:99a1:92fc:bd49 (fe80::2517:99a1:92fc:bd49) 56 data bytes

64 bytes from fe80::2517:99a1:92fc:bd49%enp0s25: icmp\_seq=1 ttl=64 time=0.574 ms

64 bytes from fe80::2517:99a1:92fc:bd49%enp0s25: icmp\_seq=2 ttl=64 time=0.277 ms

64 bytes from fe80::2517:99a1:92fc:bd49%enp0s25: icmp\_seq=3 ttl=64 time=0.285 ms

^C

--- fe80::2517:99a1:92fc:bd49 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2057ms

rtt min/avg/max/mdev = 0.277/0.378/0.574/0.138 ms

Όταν τρέχουμε την **ping** προς διευθύνεις link-local και προς το link-local all-nodes στην ομάδα multicast (ff02::1), πρέπει να οριστεί και η διεπαφή δικτύου μέσω ενός αναγνωριστικού ζώνης εμβέλειας, π.χ. ff02::1%enp0s25. Αν παραληφθεί, τότε εμφανίζεται το λάθος *connect: invalid argument* (*σύνδεση:μη έγκυρο όρισμα).*

Με ένα ping6 στην ff02::1 μπορούν να βρεθούν άλλοι κόμβοι IPv6 στο τοπικό δίκτυο:

**user@linuxsrv ~ $** ping6 ff02::1%enp0s25

PING ff02::1%enp0s25 (ff02::1%enp0s25) 56 data bytes

64 bytes from fe80::f2de:f1ff:fe5f:227%enp0s25: icmp\_seq=1 ttl=64 time=0.035 ms

64 bytes from fe80::2517:99a1:92fc:bd49%enp0s25: icmp\_seq=1 ttl=64 time=0.210 ms

64 bytes from fe80::7ac5:7dff:fef4:3e30%enp0s25: icmp\_seq=1 ttl=64 time=0.802 ms

64 bytes from fe80::4f4e:4a4c:4db3:2e2b%enp0s25: icmp\_seq=1 ttl=64 time=2.36 ms

64 bytes from fe80::ee88:92ff:fed2:b1db%enp0s25: icmp\_seq=1 ttl=64 time=149 ms

64 bytes from fe80::48f2:86ff:fe19:8991%enp0s25: icmp\_seq=1 ttl=64 time=206 ms

^C

--- ff02::1%enp0s25 ping statistics ---

2 packets transmitted, 2 received, +10 duplicates, 0% packet loss, time 1000ms

rtt min/avg/max/mdev = 0.035/46.958/205.974/71.304 ms

Υπενθυμίζεται ότι οι διευθύνσεις link-local μπορούν να χρησιμοποιηθούν από άλλους υπολογιστές που βρίσκονται στην ίδια σύνδεση, όπως και οι κανονικές διευθύνσεις:

**user@linuxsrv ~ $** ssh fe80::2517:99a1:92fc:bd49%enp0s25

Enter passphrase for key 'XXXXXXXXXXXX':

Last login: Sun Dec 22 19:14:02 2024 from fe80::f2de:f1ff:fe5f:227%enp3s0

[user@xxx ~]$

### Έλεγχος και επιδιόρθωση προβλημάτων δρομολόγησης

Η δρομολόγηση στα δίκτυα είναι πολύπλοκη. Η κίνηση του δικτύου μπορεί να μην συμπεριφέρεται με τον αναμενόμενο τρόπο. Θα παρουσιαστούν μερικά διαγνωστικά εργαλεία, που μπορεί να προσφέρουν πολύτιμη βοήθεια.

#### Εμφανίζοντας τον πίνακα δρομολόγησης

Για να δούμε πληροφορίες σχετικά με τη δρομολόγηση, χρησιμοποιούμε την εντολή **ip** με την επιλογή **route**:

**user@linuxsrv ~ $** ip route

default via 192.168.1.1 dev enp0s25 proto static metric 100

192.168.1.0/24 dev enp0s25 proto kernel scope link src 192.168.1.71 metric 100

10.0.0.0/8 dev enp0s20 proto kernel scope link src 10.0.0.11 metric 600

Ο παραπάνω είναι ο πίνακας δρομολόγησης του IPv4. Βλέπουμε ότι όλα τα πακέτα που προορίζονται για το δίκτυο 192.168.1.0/24 αποστέλλονται απ` ευθείας στον προορισμό τους μέσω της συσκευής enp0s25. Αντίστοιχα, όλα τα πακέτα με προορισμό το δίκτυο 10.0.0.0/8 στέλνονται απ` ευθείας μέσω της συσκευής enp0s20. Όλα τα άλλα πακέτα στέλνονται στον προεπιλεγμένο δρομολογητή, που βρίσκεται στην 192.168.1.1 και πάλι μέσω της enp0s25.

Προσθέτοντας την επιλογή **-6**, βλέπουμε τον πίνακα δρομολόγησης για το IPv6:

**user@linuxsrv:~$** ip -6 route

::1 dev lo proto kernel metric 256 pref medium

fe80::/64 dev eno1 proto kernel metric 1024 pref medium

Βλέποντας τις διαδρομές που εμφανίστηκαν και αγνοώντας τις εγγραφές «unreachable», που δεν υπάρχει διαδρομή για να φτάσουν εκεί πακέτα, έχουμε τις παρακάτω διαδρομές:

* τη διαδρομή για το δίκτυο 2001::/64, που χρησιμοποιεί την διεπαφή eno1 και η οποία είναι πολύ πιθανό να έχει μια διεύθυνση σε αυτό το δίκτυο
* τη διαδρομή για το δίκτυο fe80::/64, που χρησιμοποιεί τη διεπαφή eno1, για τη διεύθυνση link-local. Αν σε έναν υπολογιστή υπάρχουν πολλαπλές διεπαφές δικτύου, τότε υπάρχει μια διαδρομή e80::/64 προς τα έξω από τη διεπαφή για τη διεύθυνση link-local της διεπαφής.
* Μια προεπιλεγμένη διαδρομή προς όλα τα δίκτυα του Διαδικτύου IPv6 (::/0) για τα οποία δεν υπάρχει πιο εξειδικευμένη διαδρομή στον υπολογιστή. Αυτή περνάει από τον δρομολογητή στη διεύθυνση 2001:db8:0:1::ffff, μέσω της διεπαφής eno1.

#### Ανιχνεύοντας τις διαδρομές που ακολουθούν τα πακέτα

Για να ανιχνεύσουμε τη διαδρομή που ακολουθούν τα πακέτα στο δίκτυο προς ένα μακρινό προορισμό που περνάει από πολλούς δρομολογητές, χρησιμοποιούμε είτε την εντολή **traceroute**, είτε την εντολή **tracepath**. Με αυτό τον τρόπο μπορούμε να βρούμε κατά πόσο ένα πρόβλημα έχει να κάνει με κάποιον από τους δικούς μας δρομολογητές ή με κάποιον ενδιάμεσο. Και οι δύο εντολές βασίζονται σε πακέτα UDP σαν προεπιλογή. Το πρόβλημα έγκειται στο γεγονός ότι πολλά δίκτυα κόβουν πακέτα UDP και ICMP. Η **traceroute** έχει σαν επιλογές να χρησιμοποιήσει είτε το προεπιλεγμένο UDP, είτε το ICMP (**-I**), είτε το TCP (-**T**) για να ανιχνεύσει τη διαδρομή, όμως συνήθως η **traceroute** δεν είναι προεγκατεστημένη.

**[user@linuxsrv ~]$** tracepath access.redhat.com

1?: [LOCALHOST] pmtu 1500

1: \_gateway 0.458ms

1: \_gateway 0.682ms

2: \_gateway 0.654ms pmtu 1492

2: lo500.bbras-kln-01.forthnet.gr 8.003ms

3: he0-2-0-4.core-kln-06.forthnet.gr 7.332ms

4: be43.bgw-kln-02.forthnet.gr 7.534ms

5: no reply

6: a2-22-245-56.deploy.static.akamaitechnologies.com 6.505ms reached

Resume: pmtu 1492 hops 6 back 6

Κάθε γραμμή εξόδου αντιστοιχεί σε ένα δρομολογητή (router) ή, αλλιώς, ένα *hop*, που συναντάει το πακέτο στο δρόμο του προς τον τελικό του προορισμό. Αν είναι διαθέσιμες, εμφανίζονται και επιπλέον πληροφορίες, όπως ο *χρόνος πλήρους διαδρομής* (περιλαμβάνει και την επιστροφή στην πηγή) – *round trip timing (RTT)* και πιθανές αλλαγές στο μέγεθος της *μέγιστης μονάδας μετάδοσης – maximum transmission unit (MTU)*, που φαίνεται στο παραπάνω παράδειγμα (pmtu 1492). Αν κάπου εμφανιστεί η ένδειξη asymm, σημαίνει ότι τοι πακέτο έφτασε στον συγκεκριμένο δρομολογητή που εμφάνισε την ένδειξη και επέστρεψε από αυτόν από άλλη διαδρομή, εξ ου και το *asymmetric – ασύμμετρη*. Οι δρομολογητές που χρησιμοποιούνται για την επιστροφή δεν εμφανίζονται.

Οι αντίστοιχες εντολές για το IPv6 είναι οι **tracepath6** και **traceroute -6**:

**user@linuxsrv ~ $** tracepath6 access.redhat.com

1?: [LOCALHOST] 0.006ms pmtu 1500

1: 2a02:2149:8a6a:df00:7ac5:7dff:fef4:3e30 0.607ms

1: 2a02:2149:8a6a:df00:7ac5:7dff:fef4:3e30 0.696ms

2: 2a02:2149:8a6a:df00:7ac5:7dff:fef4:3e30 0.545ms pmtu 1492

2: no reply

3: he0-2-0-4.core-kln-06.forthnet.gr 7.814ms

4: be33.bgw-kln-02.forthnet.gr 8.023ms

5: no reply

6: g2a02-26f0-c000-0000-0000-0000-0215-4592.deploy.static.akamaitechnologies.com 6.338ms reached

Resume: pmtu 1492 hops 6 back 6

### ΈΛΕΓΧΟΣ ΚΑΙ ΕΠΙΔΙΟΡΘΩΣΗ ΣΕ ΘΥΡΕΣ ΚΑΙ ΥΠΗΡΕΣΙΕΣ

Οι υπηρεσίες TCP χρησιμοποιούν *υποδοχές – sockets* σαν τερματικά σημεία για την επικοινωνία. Οι υποδοχές είναι η σύνθεση μιας τριάδας πραγμάτων: μιας διεύθυνσης IP, ενός πρωτοκόλλου και ενός αριθμού θύρας. Οι υπηρεσίες τυπικά ακούνε σε καθορισμένες θύρες, ενώ οι πελάτες χρησιμοποιούν μια τυχαία διαθέσιμη θύρα. Τα λεγόμενα ευρέως γνωστά (well-known) ονόματα για τις καθορισμένες θύρες βρίσκονται στο αρχείο **/etc/services**.

Η εντολή **ss** μας δίνει στατιστικά για τις υποδοχές. Είναι ένα νεότερο εργαλείο που αντικαθιστά το παλιότερο **netstsat**, που είναι μέρος του πακέτου *net-tools*. Το net-tools ίσως να είναι πιο γνωστό σε διαχειριστές συστημάτων, αλλά είναι μέρος των προαιρετικών πακέτων, που δεν εγκαθίστανται πάντα.

**user@linuxsrv:~$** ss -ta

State Recv-Q Send-Q Local Address:Port Peer Address:Port Process

LISTEN 0 64 0.0.0.0:nfs 0.0.0.0:\*

LISTEN 0 128 0.0.0.0:ssh 0.0.0.0:\*

LISTEN 0 4096 0.0.0.0:sunrpc 0.0.0.0:\*

LISTEN 0 100 127.0.0.1:smtp 0.0.0.0:\*

ESTAB 0 80 192.168.0.14:ssh 192.168.0.3:44250

LISTEN 0 64 [::]:nfs [::]:\*

LISTEN 0 128 [::]:ssh [::]:\*

LISTEN 0 128 [::1]:smtp [::]:\*

LISTEN 0 4096 [::]:sunrpc [::]:\*

Στο παραπάνω παράδειγμα, οι διευθύνσεις 0.0.0.0 αντιπροσωπεύουν όλες τις διευθύνσεις IP, άρα η θύρα που χρησιμοποιείται για το ssh (στη δεύτερη γραμμή της εξόδου της εντολής **ss**)ακούει σε όλες τις διευθύνσεις IP.

Στη θέση του 0.0.0.0 μπορεί να εμφανίζεται το «\*».

Στην τέταρτη γραμμή βλέπουμε ότι η θύρα που χρησιμοποιείται για το SMTP ακούει στη διεπαφή 127.0.0.1, που είναι η διεπαφή loopback του IPv4.

Στην πέμπτη γραμμή βλέπουμε ότι υπάρχει μια ενεργή σύνδεση SSH (όπως δείχνει το «ESTAB» αριστερά στην αρχή της γραμμής) στη διεπαφή 192.168.0.14, η οποία προέρχεται από ένα σύστημα με διεύθυνση 192.168.0.3.

Στην έβδομη γραμμή βλέπουμε ότι και για το IPv6 η θύρα του SSH ακούει σε όλες τις διευθύνσεις IPv6. Εδώ το «::» αντιπροσωπεύει όλες τις διεπαφές IPv6.

Τέλος, στην όγδοη γραμμή, βλέπουμε ότι η θύρα του SMTP ακούει στη διεπαφή loopback του IPv6 (::1).

Να σημειώσουμε εδώ ότι υπάρχουν μικρές παραλλαγές στις εντολές **traceroute** και  **tracepath.** Σε κάποια συστήματα μπορεί να βρείτε ότι η εντολή **tracepath** για το IPv6 είναι η **tracepath6**, ενώ σε άλλα αυτό γίνεται με την επιλογή γραμμής **-6**, οπότε η εντολή γίνεται **tracepath -6**. Αντίστοιχα ισχύουν και για την **tracetoute**. Τέλος, αν υπάρχει π.χ. η εντολή **tracepath6**, είναι πιθανό να υπάρχει και η **tracepath4**, ώστε να είναι πιο ξεκάθαρο για ποιά έκδοση του πρωτοκόλλου IP προορίζεται η εντολή.

Ας δούμε τώρα μερικές από τις περισσότερο χρησιμοποιούμενες επιλογές γραμμής για την **ss** και **netstat**:

|  |  |
| --- | --- |
| Επιλογή | Περιγραφή |
| -n | Εμφανίζει αριθμούς, αντί για ονόματα διεπαφών και θυρών. |
| -t | Εμφανίζει εσοχές (sockets) TCP. |
| -u | Εμφανίζει εσοχές (sockets) UDP. |
| -l | Εμφανίζει μόνο εσοχές που ακούν. |
| -a | Εμφανίζει όλες τις εσοχές, και αυτές που ακούν (listening), και αυτές στις οποίες έχουν δημιοργηθεί συνδέσεις (established). |
| -p | Εμφανίζει τη διεργασία που χρησιμοποεί τις εσοχές. |
| -A inet | Εμφανίζει ενεργές συνδέσεις για την οικογένεια διευθύνσεων inet, εξαιρώντας τις εσοχές που ακούνε. Οι εξαιρούμενες είναι οι τοπικές εσοχές Unix domain. Για το **ss**, πολλές φορές εμφανίζονται τόσο οι συνδέσεις IPv4, όσο και οι IPv6 (βλ. και σημείωση παραπάνω). Για το **netstat**, με την **netstat -A inet6** εμφανίζει συνδέσεις IPv6, ενώ με τη **netstat -46** εμφανίζει συνδέσεις IPv6 και IPv4 ταυτόχρονα (και πάλι παραπέμπουμε στη σημείωση παραπάνω). |

## ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ

**Σκοπός:** Σε αυτή την άσκηση θα επιθεωρήσετε τις ρυθμίσεις δικτύου ενός εξυπηρετητή.

**Προσδοκώμενα αποτελέσματα:** Θα βρείτε και θα αναγνωρίσετε τις τρέχουσες διεπαφές δικτύου και τις βασικές διευθύνσεις δικτύου.

1. Πριν ξεκινήσετε:  
   Εισέρχεστε στον εξυπηρετητή που θα χρησιμοποιηθεί για την άσκηση χρησιμοποιώντας τα διαπιστευτήρια (όνομα χρήστη και συνθηματικό) που σας έχουν δοθεί.
2. Βρείτε τις διεπαφές δικτύου και τις διευθύνσεις MAC που τους αντιστοιχούν.  
   **student@linuxsrva ~ $** ip link

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000

link/ether 46:40:6f:89:e2:4c brd ff:ff:ff:ff:ff:ff permaddr 08:11:96:01:39:88

1. Εμφανίστε τις τρέχουσες διευθύνσεις IP και μάσκες δικτύου για όλες τις διεπαφές δικτύου.

**student@linuxsrva ~ $** ip addr

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 brd 127.255.255.255 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host proto kernel\_lo

valid\_lft forever preferred\_lft forever

2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.71/24 brd 192.168.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet6 2a02:2149:8a6a:df00:f2de:f1ff:fe5f:227/64 scope global dynamic mngtmpaddr proto kernel\_ra

valid\_lft 4294931194sec preferred\_lft 4294931194sec

inet6 fe80::f2de:f1ff:fe5f:227/64 scope link proto kernel\_ll

valid\_lft forever preferred\_lft forever

3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000

link/ether 4e:4c:e9:56:83:48 brd ff:ff:ff:ff:ff:ff permaddr 08:11:96:01:39:88

1. Εμφανίστε στατιστικά στοιχεία για τη διεπαφή enp025.

**student@linuxsrva ~ $** ip -s link show enp0s25

3: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

RX: bytes packets errors dropped missed mcast

1980510295 1370505 0 0 0 1885

TX: bytes packets errors dropped carrier collsns

69218983 610293 0 0 0 0

1. Εμφανίστε τις πληροφορίες δρομολόγησης.

**student@linuxsrva ~ $** ip route

default via 192.168.1.1 dev enp0s25 metric 3

192.168.1.0/24 dev enp0s25 proto kernel scope link src 192.168.1.71 metric 100

1. Επιβεβαιώστε ότι ο δρομολογητής είναι προσβάσιμος.

**student@linuxsrva ~ $** ping 192.168.1.1

PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.

64 bytes from 192.168.1.1: icmp\_seq=1 ttl=64 time=0.527 ms

64 bytes from 192.168.1.1: icmp\_seq=2 ttl=64 time=0.508 ms

64 bytes from 192.168.1.1: icmp\_seq=3 ttl=64 time=0.435 ms

^C

--- 192.168.1.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2065ms

rtt min/avg/max/mdev = 0.435/0.490/0.527/0.039 ms

1. Εμφανίστε όλες τις μεταβάσεις (hops) μεταξύ του τοπικού υπολογιστή και του target.example.com.

**student@linuxsrva ~ $** tracepath target.example.com

1?: [LOCALHOST] pmtu 1500

1: target.example.com 0.503ms reached

1: target.example.com 0.411ms reached

Resume: pmtu 1500 hops 1 back 2

1. Εμφανίστε όλες τις υποδοχές (sockets) TCP που ακούνε στον τοπικό υπολογιστή.

**student@linuxsrva ~ $** ss -lt

State Recv-Q Send-Q Local Address:Port Peer Address:Port

LISTEN 0 0 127.0.0.1:41795 0.0.0.0:\*

LISTEN 0 0 0.0.0.0:ssh 0.0.0.0:\*

LISTEN 0 0 \*:ssh \*:\*

1. Αποσυνδεθείτε από τον εξυπηρετητή.

**student@linuxsrva ~ $** exit

logout

Connection to linuxsrva closed.

**student@localpc ~ $**

### Ολοκλήρωση

Με την αποσύνδεση από τον εξυπηρετητή ολοκληρώθηκε η καθοδηγούμενη άσκηση.

## ΡΥΘΜΙΣΗ ΤΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΗ ΓΡΑΜΜΗ ΕΝΤΟΛΩΝ

Με την ολοκλήρωση αυτής της ενότητας, θα μπορείτε να διαχειριστείτε τις ρυθμίσεις και τις συσκευές δικτύου με τη χρήση της εντολής **nmcli**.

### ΕΝΝΟΙΕΣ ΤΟΥ NETWORKMANAGER

Ο NetworkManager είναι ένας daemon που επιβλέπει και διαχειρίζεται ρυθμίσεις δικτύου. Εκτός από τον daemon, υπάρχει και εφαρμογίδιο (applet) για την περιοχή ειδοποιήσεων του GNOME (GNOME Notification Area), που παρέχει πληροφορίες κατάστασης του δικτύου σε παραθυρικά περιβάλλοντα. Υπάρχουν τόσο εργαλεία γραμμής εντολών, όσο και γραφικά εργαλεία που επικοινωνούν με τον NetworkManager και σώζουν αρχεία ρυθμίσεων στον κατάλογο **/etc/sysconfig/network-scripts** (ο κατάλογος μπορεί να είναι διαφορετικός σε συστήματα διαφορετικών διανομών).

Για τις ανάγκες της τρέχουσας ενότητας, όταν αναφερόμαστε σε μια *συσκευή*, εννοούμε μια διεπαφή δικτύου, ενώ όταν μιλάμε για μια *σύνδεση*, εννοούμε ένα σύνολο ρυθμίσεων που μπορούν να ρυθμιστούν για μια συσκευή.

Έτσι, μόνο μία σύνδεση μπορεί να είναι *ενεργή* κάθε στιγμή για όποια συσκευή. Πολλαπλές συνδέσεις μπορεί να υπάρχουν είτε για χρήση από διαφορετικές συσκευές, είτε για να αλλάξει η ρύθμιση της ίδιας συσκευής. Π.χ., για να αλλάξει κανείς προσωρινά τις ρυθμίσεις δικτύου, αντί να αλλάξει τις ρυθμίσεις μιας σύνδεσης, απλώς αλλάζει την ενεργή σύνδεση μιας συσκευής. Αυτό γίνεται εύκολα αντιληπτό στην περίπτωση μιας συσκευής για ασύρματο δίκτυο σε έναν φορητό υπολογιστή, όπου μπορεί να χρησιμοποιούνται διαφορετικές συνδέσεις για το ασύρματο δίκτυο στο χώρο εργασίας, για το ασύρματο δίκτυο του σπιτιού και για δημόσια ασύρματα δίκτυα.

Κάθε σύνδεση έχει ένα *όνομα* ή αναγνωριστικό που λειτουργεί σαν ταυτότητά της. Για τη δημιουργία και την επεξεργασία αρχείων συνδέσεων από τη γραμμή εντολής χρησιμοποιείται το εργαλείο **nmcli.**

### ΕΜΦΑΝΙΖΟΝΤΑΣ ΠΛΗΡΟΦΟΡΙΕΣ ΔΙΚΤΥΟΥ

Η κατάσταση όλων των συσκευών δικτύου εμφανίζεται με την εντολή **nmcli dev status**:

**user@linuxsrv ~ $** nmcli dev status

DEVICE TYPE STATE CONNECTION

enp0s25 ethernet connected eth0

lo loopback connected (externally) lo

F8:1F:32:51:26:D6 bt disconnected --

ppp0 ppp disconnected --

wlp3s0 wifi disconnected --

Για να εμφανιστεί η κατάσταση όλων των συνδέσεων, δίνουμε την εντολή **nmcli con show**. Αν θέλουμε να εμφανιστούν μόνο οι ενεργές συνδέσεις, προσθέτουμε την επιλογή –**active**:

**user@linuxsrv ~ $** nmcli con show

NAME UUID TYPE DEVICE

eth0-cl be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet enp0s25

lo 39fcdab6-4401-4e3d-943f-87fc12da055d loopback lo

Brascos Hotel 551b6fb2-9619-4b54-a83b-8bd6036f8b2f wifi --

COSMOTE-22CF90 20574b2a-7535-4184-a1ee-f4fb84b2a927 wifi --

COSMOTE-361048 dba40897-0bc0-4be2-8e46-84c970fe7cd6 wifi --

Eth openNIC e5d6ccf1-0a89-4098-b477-645c4ac2e46e ethernet --

Minbar bbd38281-5e8a-42ec-9244-27f5f547e31a wifi --

OldAlliance bd76ea80-5cbc-414b-8b91-a15b2a1d803c wifi --

enp0s25 c83ddf91-b5b2-47e7-a44c-7a3acd2bd45e ethernet --

wlan0 76e438ab-f8d7-4fe6-8c50-b2eea5f5d111 wifi --

agio@AxisCity ~ $ nmcli con show --active

NAME UUID TYPE DEVICE

eth0-cloudflare be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet enp0s25

lo 39fcdab6-4401-4e3d-943f-87fc12da055d loopback lo

### ΠΡΟΣΘΕΤΟΝΤΑΣ ΜΙΑ ΝΕΑ ΣΥΝΔΕΣΗ

Για την προσθήκη νέων συνδέσεων δικτύου χρησιμοποιούμε την εντολή  **nmcli con add**. Προϋπόθεση είναι να μη χρησιμοποιείται ήδη το όνομα σε ήδη υπάρχουσα σύνδεση.

Για να προσθέσουμε μια νέα σύνδεση με το όνομα eth1 για τη διεπαφή enp0s25, η οποία θα λάβει ρυθμίσεις για το IPv4 μέσω DHCP και θέλουμε να συνδέεται αυτόματα με την εκκίνηση, δίνουμε την εντολή:  
root@linuxsrv:~# nmcli con add con-name eth1 type ethernet ifname enp0s25

Το όνομα της σύνδεσης δίνεται με την επιλογή **con-name**, ενώ η διεπαφή για την οποία προορίζεται με την επιλογή **ifname**. Με την εντολή διασφαλίζεται και το να πάρει ρυθμίσεις δικτύου IPv6, με το να ακούει για advertisements δρομολογητών στην τοπική σύνδεση. Το όνομα του αρχείου ρυθμίσεων βασίζεται στο όνομα που δώσαμε στη σύνδεση και σώζεται στο αρχείο **/etc/sysconfig/network-scripts/ifcfg-eth1**.

Για να δημιουργήσουμε μια σύνδεση με το ίδιο όνομα (eth1) για τη διεπαφή enp0s25, αλλά με στατική διεύθυνση IPv4 και πρόθεμα δικτύου 192.168.1.5/24 και προκαθορισμένη πύλη εξόδου (default gateway) την 192.168.1.254 και αυτόματη σύνδεση κατά την εκκίνηση, δίνουμε την εντολή:

**root@linuxsrv:~#** nmcli con add con-name eth1 type ethernet ifname enp0s25 \

> ip4 192.168.1.5/24 gw4 192.168.1.254

Επειδή η εντολή είναι μεγάλη και δεν χωράει σε μια γραμμή του τερματικού, σαν τελευταίο χαρακτήρα στη γραμμή βάλαμε τον χαρακτήρα διαφυγής φλοιού «\» και συνεχίσαμε την εντολή στην επόμενη γραμμή μετά το «>» που μας έβγαλε ο φλοιός.

Το όνομα του αρχείου ρυθμίσεων είναι το ίδιο με του προηγούμενου παραδείγματος.

Τέλος, ας δημιουργήσουμε και μια σύνδεση με το ίδιο όνομα (eth1) για τη διεπαφή enp0s25, με στατικές διευθύνσεις IPv4 και IPv6, δίνοντας τη διεύθυνση IPv6 και πρόθεμα δικτύου 2001:deaf:a:8::b000:1009/64, προκαθορισμένη πύλη εξόδου IPv6 την 2001:deaf:a:8::1, ενώ για το IPv4 διεύθυνση και πρόθεμα δικτύου 192.3.5.6/24 και προκαθορισμένη πύλη εξόδου την 192.3.5.1, πάλι με αυτόματη σύνδεση κατά την εκκίνηση. Η ρύθμιση πάλι σώζεται στο αρχείο **/etc/sysconfig/network-scripts/ifcfg-eno2**. Η εντολή είναι η παρακάτω, πάλι με χαρακτήρες διαφυγής φλοιού:

**root@linuxsrv:~#** nmcli con add con-name eth1 type ethernet ifname enp0s25 \

> ip6 2001:deaf:a:8::b000:1009/64 gw6 2001:deaf:a:8::1 ip4 192.3.5.6/24 \

> gw4 192.3.5.1

### ΕΛΕΓΧΟΝΤΑΣ ΤΙΣ ΣΥΝΔΕΣΕΙΣ ΔΙΚΤΥΟΥ

Με την εντολή **nmcli con up *όνομα*** ενεργοποιείται η σύνδεση *όνομα* στη διεπαφή δικτύου με την οποία σχετίζεται. Το  *όνομα* είναι το όνομα μιας *σύνδεσης* και όχι μιας διεπαφής δικτύου.

**root@linuxsrv:~#** nmcli con up ifcfg-eth1

Αν θέλουμε να αποσυνδέσουμε μια *συσκευή* δικτύου (και όχι σύνδεση) και να την κατεβάσουμε (= απενεργοποιήσουμε), δίνουμε την εντολή **nmcli dev disconnect *συσκευή***. Το «disconnect» μπορεί να συντομευτεί σε «dis»:

**root@linuxsrv:~#** nmcli dev dis enp0s25

Η παραπάνω εντολή είναι ο ενδεδειγμένος τρόπος για να απενεργοποιήσουμε μια διεπαφή δικτύου. Η αντίστοιχη εντολή για απενεργοποίηση σύνδεσης (**nmcli con down *όνομα***) δεν ενδείκνυται για να απενεργοποιήσουμε μια διεπαφή δικτύου, επειδή κατεβάζει μεν τη σύνδεση, αλλά η προκαθορισμένη συμπεριφορά στις περισσότερες ενσύρματες συνδέσεις είναι να έχουν ενεργοποιημένη την αυτόματη σύνδεση (**autoconnect**), με αποτέλεσμα να ενεργοποιείται η σύνδεση μόλις είναι διαθέσιμη η διεπαφή δικτύου με την οποία σχετίζεται. Αυτό έχει σαν αποτέλεσμα, μόλις η **nmcli con down *όνομα*** κατεβάσει τη διεπαφή, το NetworkManager να την ξανανεβάζει, εκτός κι αν η σύνδεση (καλώδιο δικτύου) αποσυνδεθεί εντελώς από τη διεπαφή.

### ΑΛΛΑΖΟΝΤΑΣ ΤΙΣ ΡΥΘΜΙΣΕΙΣ ΣΥΝΔΕΣΕΩΝ ΔΙΚΤΥΟΥ

Οι συνδέσεις του NetworkManager έχουν δύο ειδών ρυθμίσεις. Πρώτα υπάρχουν οι *στατικές* ιδιότητες των ρυθμίσεων, που καθορίζονται από τον/ην διαχειριστή/ρια και που φυλάγονται σε αρχεία ρυθμίσεων στο **/etc/sysconfig/network-scripts/ifcfg-\***. Υπάρχουν επίσης και τα *ενεργά* δεδομένα σύνδεσης, τα οποία λαμβάνονται από έναν εξυπηρετητή DHCP και τα οποία φυλάγονται προσωρινά.

Για να εμφανιστούν οι τρέχουσες ρυθμίσεις για μια σύνδεση, τρέχουμε την εντολή **nmcli con show *όνομα***, όπου  *όνομα* το όνομα της σύνδεσης. Η έξοδος που παράγεται είναι αρκετά μεγάλη. Οι ρυθμίσεις που είναι με πεζά είναι οι στατικές ιδιότητες που αναφέρθηκε παραπάνω, ενώ με κεφαλαία εμφανίζονται οι ενεργές ρυθμίσεις που χρησιμοποιούνται προσωρινά για την τρέχουσα ενεργοποιημένη σύνδεση.

**user@linuxsrv ~ $** nmcli con show eth0-cl

connection.id: eth0-cl

connection.uuid: be65d113-2ed6-425a-a867-6a6704a8ad37

connection.stable-id: --

connection.type: 802-3-ethernet

connection.interface-name: enp0s25

connection.autoconnect: yes

connection.autoconnect-priority: 5

connection.autoconnect-retries: -1 (default)

connection.multi-connect: 0 (default)

connection.auth-retries: -1

connection.timestamp: 1735647289

connection.permissions: --

connection.zone: --

connection.controller: --

connection.master: --

connection.slave-type: --

connection.port-type: --

connection.autoconnect-slaves: -1 (default)

connection.autoconnect-ports: -1 (default)

connection.down-on-poweroff: -1 (default)

connection.secondaries: --

connection.gateway-ping-timeout: 0

connection.metered: unknown

connection.lldp: default

connection.mdns: -1 (default)

connection.llmnr: -1 (default)

connection.dns-over-tls: -1 (default)

connection.mptcp-flags: 0x0 (default)

connection.wait-device-timeout: -1

connection.wait-activation-delay: -1

802-3-ethernet.port: --

802-3-ethernet.speed: 0

802-3-ethernet.duplex: --

802-3-ethernet.auto-negotiate: no

802-3-ethernet.mac-address: F0:DE:F1:5F:02:27

802-3-ethernet.cloned-mac-address: --

802-3-ethernet.generate-mac-address-mask:--

802-3-ethernet.mac-address-denylist: --

802-3-ethernet.mtu: auto

802-3-ethernet.s390-subchannels: --

802-3-ethernet.s390-nettype: --

802-3-ethernet.s390-options: --

802-3-ethernet.wake-on-lan: default

802-3-ethernet.wake-on-lan-password: --

802-3-ethernet.accept-all-mac-addresses:-1 (default)

ipv4.method: manual

ipv4.dns: 1.1.1.1,1.0.0.1

( ... )

Για να αλλάξουμε τις ρυθμίσεις μιας σύνδεσης, χρησιμοποιούμε την εντολή **nmcli con mod *όνομα***. Οι αλλαγές σώζονται και στο αρχείο της σύνδεσης **/etc/sysconfig/network-scripts/ifcfg-*όνομα***. Για να δείτε όλες τις διαθέσιμες ρυθμίσεις, μπορείτε να ανατρέξετε στη σελίδα man του nm-settings(5). Εκεί θα δείτε ότι υπάρχουν και ρυθμίσεις που μπορούν να πάρουν πολλαπλές τιμές, οπότε μπορούν να προστεθούν ή να αφαιρεθούν τιμές από τη λίστα τιμών με την προσθήκη ενός συμβόλου + ή – ακριβώς μπροστά από το όνομα της ρύθμισης.

Για να αλλάξουμε τη διεύθυνση IPv4 σε 192.3.5.7/24 και την προεπιλεγμένη πύλη εξόδου την 192.3.5.1 στη σύνδεση static-enp0s20, δίνουμε την εντολή:

**root@linuxsrv:~#** nmcli con mod static-enp0s20 ipv4.addresses \

> “192.3.5.7/24 192.3.5.1”

Αντίστοιχα, για να αλλάξουμε τη διεύθυνση IPv6 σε 2001:deaf:a:7::b000:123a/64 και την προεπιλεγμένη πύλη εξόδου σε 2001:deaf:a:7::1 στη σύνδεση static-enp0s20, δίνουμε την εντολή:

**root@linuxsrv:~#** nmcli con mod static-enp0s20 ipv6.addresses \

> “2001:deaf:a:7::b000:123a/64 2001:deaf:a:7::1”

Να επισημάνουμε εδώ ότι τώρα η σύνδεση παίρνει τις ρυθμίσεις (όπως διεύθυνση IP κλπ.) μέσω του πρωτοκόλλου DHCP από κάποιον εξυπηρετητή DHCP. Αν θέλουμε να αλλάξουμε τις ρυθμίσεις μιας σύνδεσης, ώστε να παίρνει τις πληροφορίες σύνδεσης μόνο από αρχείο στατικών ρυθμίσεων (όπως στα παραπάνω παραδείγματα), τότε πρέπει να αλλάξουμε και τη ρύθμιση **ipv4.method** από auto σε manual.

Αντίστοιχα για το IPv6, αν η σύνδεση παίρνει τις πληροφορίες μέσω του SLAAC ή από έναν εξυπηρετητή DHCPv6 και με την αλλαγή θα τις παίρνει μόνο από αρχείο ρυθμίσεων, πρέπει η ρύθμιση **ipv6.method** να αλλάξει από auto σε manual.

Αν δεν γίνει αυτό, η σύνδεση μπορεί να κολλήσει ή να μην ολοκληρωθεί επιτυχώς όταν ενεργοποιηθεί, ή να πάρει διεύθυνση IP από έναν εξυπηρετητή DHCP, DHCPv6 ή μέσω SLAAC επιπλέον της στατικής διεύθυνσης.

### ΑΦΑΙΡΩΝΤΑΣ ΜΙΑ ΣΥΝΔΕΣΗ ΔΙΚΤΥΟΥ

Για να διαγράψουμε μια σύνδεση που ονομάζεται *όνομα* δίνουμε την εντολή **nmcli con del *όνομα***. Η εντολή την αφαιρεί από το σύστημα, την αποσυνδέει από τη συσκευή και διαγράφεται το αρχείο **/etc/sysconfig/network-scripts/ifcfg-*όνομα***.

### ΔΙΚΑΙΩΜΑΤΑ ΑΛΛΑΓΩΝ ΣΤΙΣ ΡΥΘΜΙΣΕΙΣ ΔΙΚΤΥΟΥ

Οποιεσδήποτε απαραίτητες αλλαγές στις ρυθμίσεις δικτύου μπορούν να γίνουν από το χρήστη root. Επιπλέον, αν ο υπολογιστής είναι σταθμός εργασίας ή φορητός, τότε στο άτομο που έχει συνδεθεί άμεσα μέσω του πληκτρολογίου, είτε σε γραφικό περιβάλλον, είτε σε κονσόλα κειμένου, συνήθως του επιτρέπεται να αλλάζει τις ρυθμίσεις δικτύου, χωρίς να έχει δικαιώματα διαχειριστή. Αυτό επειδή θεωρείται ότι οι χρήστες/χρήστριες του υπολογιστή μπορεί να χρειαστεί να ρυθμίσουν, να ενεργοποιήσουν ή να απενεργοποιήσουν τις ενσύρματες ή ασύρματες διεπαφές, ανάλογα με τις ανάγκες της στιγμής.

Αντίθετα, στην περίπτωση εξυπηρετητή, κανονικά οι μόνοι/ες χρήστες/χρήστριες που συνδέονται τοπικά στο μηχάνημα πρέπει να είναι διαχειριστές/ριες.

Οι κανονικοί χρήστες που συνδέονται μέσω **ssh** δεν έχουν δικαιώματα να αλλάξουν τις ρυθμίσεις δικτύου χωρίς να γίνουν root.

Για να δείτε τι δικαιώματα έχετε, μπορείτε να δώσετε την εντολή **nmcli gen permissions**.

### ΣΥΝΟΨΗ ΤΩΝ ΕΝΤΟΛΩΝ

|  |  |
| --- | --- |
| ΕΝΤΟΛΗ | ΣΚΟΠΟΣ |
| nmcli dev status | Εμφανίζει την κατάσταση όλων των διεπαφών δικτύου σύμφωνα με το NetworkManager. |
| nmcli con show | Εμφανίζει όλες τις συνδέσεις σε μορφή λίστας. |
| nmcli con show *όνομα* | Εμφανίζει τις τρέχουσες ρυθμίσεις για τη σύνδεση *όνομα.* |
| nmcli con add con-name *όνομα* | Προσθέτει μια νέα σύνδεση με την ονομασία *όνομα.* |
| nmcli con mod *όνομα* | Αλλάζει τη σύνδεση *όνομα.* |
| nmcli con reload | Ξαναφορτώνει τα αρχεία ρυθμίσεων (χρήσιμο για περιπτώσεις χειροκίνητων αλλαγών). |
| nmcli con up *όνομα* | Ενεργοποιεί τη σύνδεση *όνομα.* |
| nmcli dev dis *συσκευή* | Απεπεργοποιεί και αποσυνδέει την τρέχουσα σύνδεση στη διεπαφή δικτύου *συσκευή.* |
| nmcli con del *όνομα* | Αφαιρεί τη σύνδεση *όνομα* και το αρχείο ρυθμίσεών της. |

## ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ

### ΡΥΘΜΙΣΗ ΤΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΗ ΓΡΑΜΜΗ ΕΝΤΟΛΗΣ

Σε αυτή την άσκηση θα διαμορφώσετε τις ρυθμίσεις του δικτύου με χρήση της **nmcli.**

1. Αποτελέσματα: Θα πρέπει να μπορέσετε να μετατρέψετε ένα σύστημα από DHCP σε στατική ρύθμιση.
2. Πριν ξεκινήσετε: Εισέρχεστε στον εξυπηρετητή που θα χρησιμοποιηθεί για την άσκηση χρησιμοποιώντας τα διαπιστευτήρια (όνομα χρήστη και συνθηματικό) που σας έχουν δοθεί.
3. Βρείτε τις διεπαφές δικτύου και τις διευθύνσεις MAC που τους αντιστοιχούν. Μας ενδιαφέρει η διεπαφή enp0s25 και η διεύθυνση MAC που έχει.

**student@linuxsrva ~ $** ip link

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: **enp0s25**: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000

link/ether **f0:de:f1:5f:02:27** brd ff:ff:ff:ff:ff:ff

3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000

link/ether 46:40:6f:89:e2:4c brd ff:ff:ff:ff:ff:ff permaddr 08:11:96:01:39:88

* 1. Εμφανίστε όλες τις συνδέσεις δικτύου χρησιμοποιώντας την **nmcli**.

**student@linuxsrva ~ $** nmcli con show

NAME UUID TYPE DEVICE

enp0s25 be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet enp0s25

eth0 c83ddf91-b5b2-47e7-a44c-7a3acd2bd45e ethernet --

* 1. Εμφανίστε μόνο την ενεργή σύνδεση.   
     Το όνομα της διασύνδεσης δικτύου πρέπει να εμφανιστεί κάτω από την ετικέτα DEVICE, ενώ το όνομα της ενεργής σύνδεσης για τη συγκεκριμένη συσκευή θα είναι στην ίδια γραμμή, κάτω από την επικεφαλίδα NAME.

**student@linuxsrva ~ $** nmcli con show --active

NAME UUID TYPE DEVICE

enp0s25 be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet enp0s25

* 1. Εμφανίστε όλες τις ρυθμίσεις για την ενεργή σύνδεση.

**student@linuxsrva ~ $** nmcli con show "enp0s25"

connection.id: enp0s25

connection.uuid: be65d113-2ed6-425a-a867-6a6704a8ad37

connection.stable-id: --

connection.type: 802-3-ethernet

connection.interface-name: enp0s25

connection.autoconnect: yes

*[ … ]*

ipv4.method: manual

ipv4.dns: 193.92.150.1,193.92.3.10,192.168.1.1

ipv4.dns-search: --

ipv4.dns-options: --

ipv4.dns-priority: 100

ipv4.addresses: 192.168.1.71/24

ipv4.gateway: 192.168.1.1

*[ … ]*

(Τα αποτελέσματα εμφανίζονται μέσω προγράμματος pager, οπότε με το κενό πηγαίνετε στην επόμενη οθόνη και πατώντας το πλήκτρο **q** βγαίνετε από την εντολή.)

* 1. Εμφανίστε την κατάσταση της συσκευής.

**student@linuxsrva ~ $** nmcli dev status

DEVICE TYPE STATE CONNECTION

enp0s25 ethernet connected eth0-cloudflare

lo loopback connected (externally) lo

* 1. Εμφανίστε τις ρυθμίσεις για τη συσκευή enp0s25.

**student@linuxsrva ~ $** nmcli dev show enp0s25

GENERAL.DEVICE: enp0s25

GENERAL.TYPE: ethernet

GENERAL.HWADDR: F0:DE:F1:5F:02:27

GENERAL.MTU: 1500

GENERAL.STATE: 100 (connected)

GENERAL.CONNECTION: eth0-cloudflare

GENERAL.CON-PATH: /org/freedesktop/NetworkManager/ActiveConnection/2

WIRED-PROPERTIES.CARRIER: on

IP4.ADDRESS[1]: 192.168.1.71/24

IP4.GATEWAY: 192.168.1.1

IP4.ROUTE[1]: dst = 0.0.0.0/0, nh = 192.168.1.1, mt = 100

IP4.ROUTE[2]: dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100

IP4.ROUTE[3]: dst = 0.0.0.0/0, nh = 192.168.1.1, mt = 3

IP4.DNS[1]: 1.1.1.1

IP4.DNS[2]: 1.0.0.1

IP6.ADDRESS[1]: 2a02:2149:8a6a:df00:f2de:f1ff:fe5f:227/64

IP6.ADDRESS[2]: fe80::f2de:f1ff:fe5f:227/64

IP6.GATEWAY: fe80::7ac5:7dff:fef4:3e30

IP6.ROUTE[1]: dst = 2a02:2149:8a6a:df00::/64, nh = ::, mt = 256

IP6.ROUTE[2]: dst = fe80::/64, nh = ::, mt = 256

IP6.ROUTE[3]: dst = ::/0, nh = fe80::7ac5:7dff:fef4:3e30, mt = 1024

1. Δημιουργείστε μια στατική σύνδεση με την ίδια διεύθυνση IPv4, το ίδιο πρόθεμα δικτύου και την ίδια προεπιλεγμένη πύλη εξόδου (default gateway).

**Προσοχή:** πρόκειται να αλλάξετε ρυθμίσεις στη δικτυακή σύνδεση που χρησιμοποιείτε και για την πρόσβασή σας στην εικονική μηχανή. Ρύθμιση με λάθος τιμές μπορεί να προκαλέσει απώλεια της πρόσβασης στη μηχανή.

**student@linuxsrva ~ $** sudo nmcli con add con-name enp0s25-static ifname enp0s25 type ethernet ip4 192.168.1.71/24 gw4 192.168.1.1

Connection ‘enp0s25-static’ (XXXX-XXXX-XXXX) successfully added.

1. Τροποποιήστε τη νέα σύνδεση, προσθέτοντας τη ρύθμιση για το DNS.

**student@linuxsrva ~ $** sudo nmcli con mod enp0s25-static ipv4.dns 1.1.1.1

1. Εμφανίστε και ενεργοποιήστε τη νέα σύνδεση.
   1. Εμφανίστε όλες τις συνδέσεις.

**student@linuxsrva ~ $** nmcli con show

NAME UUID TYPE DEVICE

enp0s25 e98403c1-0d85-4aeb-b33d-f9fc85ae8ad8 ethernet enp0s25

lo f81deb57-7238-452d-be01-5cbf448734d2 loopback lo

enp0s25-static be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet --

* 1. Εμφανίστε την ενεργή σύνδεση.

**student@linuxsrva ~ $** nmcli con show --active

enp0s25 e98403c1-0d85-4aeb-b33d-f9fc85ae8ad8 ethernet enp0s25

* 1. Ενεργοποιήστε τη νέα σύνδεση emp0s25-static.

**student@linuxsrva ~ $** nmcli con up enp0s25

Connection successfully activated (D-Bus active path: /org/freedesktop/

NetworkManager/ActiveConnection/3)

* 1. Επιβεβαιώστε ότι είναι ενεργή η νέα σύνδεση.

**student@linuxsrva ~ $** nmcli con show --active

enp0s25-static be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet enp0s25

1. Ελέγξτε τη συνδεσιμότητα με τη χρήση των νέων διευθύνσεων δικτύου.
   1. Επιβεβαιώστε τη διεύθυνση IP.

**student@linuxsrva ~ $** ip addr show enp0s25

3: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.71/24 brd 192.168.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet6 2a02:2149:8a36:1700:f2de:f1ff:fe5f:227/64 scope global dynamic mngtmpaddr proto kernel\_ra

valid\_lft 4294939482sec preferred\_lft 4294939482sec

inet6 fe80::f2de:f1ff:fe5f:227/64 scope link proto kernel\_ll

valid\_lft forever preferred\_lft forever

* 1. Επιβεβαιώστε την προεπιλεγμένη πύλη εξόδου (default gateway).

**student@linuxsrva ~ $** ip route

default via 192.168.1.1 dev enp0s25 proto static metric 100

192.168.1.0/24 dev enp0s25 proto kernel scope link src 192.168.1.71 metric 100

* 1. Κάντε ping τη διεύθυνση DNS.

**student@linuxsrva ~ $** ping -c3 192.168.1.1

PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.

64 bytes from 192.168.1.1: icmp\_seq=1 ttl=64 time=0.509 ms

64 bytes from 192.168.1.1: icmp\_seq=2 ttl=64 time=0.509 ms

64 bytes from 192.168.1.1: icmp\_seq=3 ttl=64 time=0.524 ms

--- 192.168.1.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2072ms

rtt min/avg/max/mdev = 0.509/0.514/0.524/0.007 ms

1. Ρυθμίστε την αρχική σύνδεση, ώστε να μην ενεργοποιείται αυτόματα με την εκκίνηση και επιβεβαιώστε ότι χρησιμοποιείται η στατική σύνδεση όταν επανεκκινεί το σύστημα.
   1. Απενεργοποιήστε την αυτόματη ενεργοποίηση της αρχικής σύνδεσης κατά την εκκίνηση.

**student@linuxsrva ~ $** sudo nmcli con mod enp0s25 connection.autoconnect no

* 1. Επανεκκινήστε το σύστημα.

**student@linuxsrva ~ $** sudo reboot

Connection to linuxsrva closed by remote host.

Connection to linuxsrva closed.

**student@localpc ~ $**

* 1. Αφού ξανασυνδεθείτε στο μηχάνημα, εμφανίστε την ενεργή σύνδεση.

**student@linuxsrva ~ $** nmcli con show --active

enp0s25-static be65d113-2ed6-425a-a867-6a6704a8ad37 ethernet enp0s25

* 1. Αποσυνδεθείτε από το μηχάνημα.

**student@linuxsrva ~ $** exit

logout

Connection to linuxsrva closed.

**student@localpc ~ $**

## ΕΠΕΞΕΡΓΑΣΙΑ ΑΡΧΕΙΩΝ ΡΥΘΜΙΣΕΩΝ ΔΙΚΤΥΟΥ

Σε αυτή την ενότητα περιγράφεται η τροποποίηση των ρυθμίσεων δικτύου μέσω της επεξεργασίας των αρχείων ρυθμίσεων.

### ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΑΡΧΕΙΩΝ ΡΥΘΜΙΣΕΩΝ ΣΥΝΔΕΣΕΩΝ

Οι αλλαγές που κάνουμε με την εντολή **nmcli con mod *όνομα*** σώζονται αυτόματα στο **/etc/sysconfig/network-scripts/ifcfg-*όνομα***. Αυτό το αρχείο μπορεί να το επεξεργαστεί κανείς και χειροκίνητα με έναν διορθωτή κειμένου. Σε αυτή την περίπτωση, μετά τις αλλαγές πρέπει να τρέξουμε την εντολή **nmcli con reload**, ώστε το NetworkManager να διαβάσει τις αλλαγές στις ρυθμίσεις.

Για λόγους συμβατότητας με το παρελθόν, οι ρυθμίσεις που σώζονται στο αρχείο αυτό έχουν διαφορετικά ονόματα και σύνταξη από αυτά του nm-settings(5). Στον παρακάτω πίνακα φαίνονται οι αντιστοιχίες των κυριότερων ρυθμίσεων σε αυτές του **ifcfg-\***.

|  |  |  |
| --- | --- | --- |
| nmcli con mod | Αρχείο ifcfg-\* | Αποτέλεσμα |
| ipv4.method manual | **BOOTPROTO=none** | Οι διευθύνσεις IPv4 ρυθμίζονται στατικά. |
| ipv4.method auto | **BOOTPROTO=dhcp** | Παίρνει τις ρυθμίσεις από έναν εξυπηρετητή DHCP. Αν ρυθμίζονται και στατικές διευθύνσεις, δεν ενεργοποιούνται μέχρι να ληφθούν πληροφορίες από τον εξυπηρετητή DHCP. |
| ipv4.addresses “192.3.5.6/24 192.3.5.1” | **IPADDR=192.3.5.6**  **PREFIX=24**  **GATEWAY0=192.3.5.1** | Ορίζει νέα στατική διεύθυνση IPv4, πρόθεμα δικτύου και προκαθορισμένη πύλη εξόδου. Σε περίπτωση που ορίζονται παραπάνω από μια για τη σύνδεση, τότε αντί για 0 οι ρυθμίσεις του  **ifcfg-\*** έχουν κατάληξη 1, 2, 3 κλπ. |
| ipv4.dns 1.1.1.1 | **DNS0=1.1.1.1** | Αλλάζει το αρχείο **/etc/resolv.conf**, ώστε να χρησιμοποιεί τον συγκεκριμένο εξυπηρετητή ονομάτων (nameserver). |
| ipv4.dns-search some-domain.com | **DOMAIN=some-domain.com** | Αλλάζει το αρχείο **/etc/resolv.conf**, ώστε να χρησιμοποιεί αυτό το όνομα domain στη ρύθμιση **search**. |
| ipv4.ignore-auto-dns true | **PEERDNS=no** | Να αγνοηθούν οι πληροφορίες για εξυπηρετητές dns από τον εξυπηρετητή DHCP. |
| ipv6.method manual | **IPV6\_AUTOCONF=no** | Οι διευθύνσεις IPv6 ρυθμίζονται στατικά. |
| ipv6.method auto | **IPV6\_AUTOCONF=yes** | Οι ρυθμίσεις δικτύου γίνονται με χρήση του SLAAC από τις adverisements του δρομολογητή. |
| ipv6.method dhcp | **IPV6\_AUTOCONF=no**  **DHCPV6C=yes** | Οι ρυθμίσεις δικτύου γίνονται μέσω του DHCPv6, αλλά όχι μέσω του SLAAC. |
| ipv6.addresses “2001:deaf:a:7::b000:123a/64 2001:deaf:a:7::1” | **IPV6ADDR= 2001:deaf:a:7::b000:123a/64**  **IPV6\_DEFAULTGW= 2001:deaf:a:7::1** | Ορίζει στατική διεύθυνση IPv6, πρόθεμα δικτύου και προκαθορισμένη πύλη εξόδου. Αν ορίζονται πάνω από μία διευθύνσεις για τη σύνδεση, τότε η IPV6\_SECONDARIES δέχεται μια λίστα από ορισμούς διευθύνσεων/προθεμάτων χωρισμένες με κενά, ολόκληρη μέσα σε διπλά εισαγωγικά. |
| ipv6.dns … | **DNS= …** | Αλλάζει το **/etc/resolv.conf**, ώστε να χρησιμοποιεί το συγκεκριμένο εξυπηρετητή ονομάτων, όπως ακριβώς και με το IPv4. |
| ipv6.dns-search some-domain.com | **DOMAIN=some-domain.com** | Όπως και με το IPv4, αλλάζει το αρχείο **/etc/resolv.conf**, ώστε να χρησιμοποιεί αυτό το όνομα domain στη ρύθμιση **search**. |
| ipv6.ignore-auto-dns true | **IPV6\_PEERDNS=no** | Να αγνοηθούν οι πληροφορίες για εξυπηρετητές dns από τον εξυπηρετητή DHCP. |
| connection.autoconnect yes | **ONBOOT=yes** | Αυτόματη ενεργοποίηση της συγκεκριμένης σύνδεσης κατά την εκκίνηση. |
| connection.id enp0s25 | **NAME=enp0s25** | Το όνομα της σύνδεσης. |
| connection.interface-name enp0s25 enp0s25 | **DEVICE=enp0s25** | Η σύνδεση συσχετίζεται με τη διεπαφή δικτύου που ονομάζεται έτσι. |
| 802-3-ethernet.mac-address ... | **HWADDR= ...** | Η σύνδεση συσχετίζεται με τη διεπαφή δικτύου που έχει τη συγκεκριμένη δειύθυνση MAC. |

### ΑΛΛΑΖΟΝΤΑΣ ΤΙΣ ΡΥΘΜΙΣΕΙΣ ΔΙΚΤΥΟΥ

Ένας άλλος τρόπος για να ρυθμίσουμε το δίκτυο είναι το να αλλάξουμε απευθείας τα ίδια τα αρχεία ρυθμίσεων. Θυμίζουμε ότι τα αρχεία ρυθμίσεων συνδέσεων ελέγχουν τις διεπαφές λογισμικού ξεχωριστά για κάθε μία συσκευή δικτύου και συνήθως ονομάζονται **/etc/sysconfig/network-scripts/ifcfg-*όνομα***, όπου *όνομα* το όνομα της συσκευής ή σύνδεσης που ελέγχει το αρχείο ρυθμίσεων.

Οι παρακάτω είναι οι καθιερωμένες μεταβλητές που μπορούν να βρεθούν σε ένα αρχείο στατικών ή δυναμικών ρυθμίσεων για το IPv4.

|  |  |  |
| --- | --- | --- |
| ΣΤΑΤΙΚΕΣ | ΔΥΝΑΜΙΚΕΣ | ΚΑΙ ΓΙΑ ΤΙΣ ΔΥΟ ΠΕΡΙΠΤΩΣΕΙΣ |
| BOOTPROTO=none  IPADDR0=10.1.20.177  PREFIX0=24  GATEWAY0=10.1.20.254  DEFROUTE=yes  DNS1=10.1.24.254 | **BOOTPROTO=dhcp** | **DEVICE=enp0s25**  **NAME=”static-enp0s25”**  **ONBOOT=yes**  **UUID=fac3(...)d3af**  **USERCTL=yes** |

Στις στατικές ρυθμίσεις βλέπουμε ότι οι μεταβλητές για τη διεύθυνση IP, το πρόθεμα και την πύλη έχουν έναν αριθμό στο τέλος (εδώ το 0). Με αυτό τον τρόπο μπορούμε να έχουμε πολλαπλά σύνολα τιμών που μπορούν να ανατεθούν στη διεπαφή.

Ο αριθμός που έχει η μεταβλητή για το DNS καθορίζει τη σειρά αναζήτησης σε περίπτωση που ορίζονται παραπάνω από ένας εξυπηρετητές.

Αφού ολοκληρωθούν οι αλλαγές στα αρχεία ρυθμίσεων, πρέπει να τρέξουμε την εντολή  **nmcli con reload**, ώστε το NetworkManager να διαβάσει τις αλλαγές στις ρυθμίσεις. Για να ενεργοποιηθούν οι αλλαγές χρειάζεται να γίνει επανεκκίνηση στη διεπαφή.

**root@linuxsrv:~#** nmcli con reload

**root@linuxsrv:~#** nmcli con down “static-enp0s25”

**root@linuxsrv:~#** nmcli con up “static-enp0s25”

## ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ ΕΠΕΞΕΡΓΑΣΙΑ ΑΡΧΕΙΩΝ ΡΥΘΜΙΣΕΩΝ ΔΙΚΤΥΟΥ

Σε αυτή την άσκηση θα επεξεργαστείτε χειροκίνητα (χωρίς τη χρήση ειδικών εργαλείων ρύθμισης) τα αρχεία ρυθμίσεων του δικτύου**.**

1. Αποτελέσματα:

Θα πρέπει να μπορέσετε να προσθέσετε μια επιπλέον διεύθυνση δικτύου στο σύστημα.

1. Πριν ξεκινήσετε:

Εισέρχεστε στον εξυπηρετητή που θα χρησιμοποιηθεί για την άσκηση χρησιμοποιώντας τα διαπιστευτήρια (όνομα χρήστη και συνθηματικό) που σας έχουν δοθεί.

1. Χρησιμοποιώντας την εντολή **sudo -i**, αλλάξτε στον χρήστη root.

**student@linuxsrva ~ $** sudo -i

Password:

**root@linuxsrva ~ #**

1. Στα πλαίσια αυτής της άσκησης, θεωρούμε ότι η συσκευή δικτύου έχει διεύθυνση MAC την f0:de:f1:5f:02:27 και ότι ονομάζεται enp0s25. Η ενεργή σύνδεση ονομάζεται επίσης enp0s25 (άρα το αρχείο ρυθμίσεών της είναι το **/etc/synsconfig/network-scripts/ifcfg-enp0s25**).

Εμφανίστε τις δικές σας ρυθμίσεις.

**student@linuxsrva ~ $** ip link

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

1. Σαν χρήστης root, επεξεργαστείτε το αρχείο **/etc/sysconfig/network-scripts/ifcfg-enp0s25** στο linuxsrva, προσθέτοντας μια επιπλέον διεύθυνση, την **10.0.1.1/24**.
   1. Προσθέστε στο τέλος του αρχείου την παρακάτω εγγραφή, που ορίζει τη νέα διεύθυνση:

**root@linuxsrva ~ #** echo "IPADDR1=10.0.1.1" >> \

> /etc/sysconfig/network-scripts/ifcfg-enp0s25

* 1. Προσθέστε στο τέλος του αρχείου άλλη μία εγγραφή, που καθορίζει το πρόθεμα δικτύου.

**root@linuxsrva ~ #** echo "PREFIX1=24" >> /etc/sysconfig/network-scripts/ifcfg-enp0s25

1. Ενεργοποιήστε τη νέα διεύθυνση.
   1. Επαναφορτώστε τις αλλαγές ρυθμίσεων.

**root@linuxsrva ~ #** nmcli con reload

* 1. Επανεκκινήστε τη σύνδεση με τις νέες ρυθμίσεις.

**root@linuxsrva ~ #** nmcli con up "enp0s25"

Connection successfully activated (D-Bus active path: /org/freedesktop/

NetworkManager/ActiveConnection/3)

* 1. Επιβεβαιώστε ότι δόθηκε η νέα διεύθυνση.

**root@linuxsrva ~ #** ip addr show enp0s25

3: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.71/24 brd 192.168.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet 10.0.1.1/24 brd 10.0.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet6 fe80::f2de:f1ff:fe5f:227/64 scope link proto kernel\_ll

valid\_lft forever preferred\_lft forever

* 1. Αποσυνδεθείτε από τον linuxsrva, επιστρέφοντας στον τοπικό σας υπολογιστή.

**student@linuxsrva ~ $** exit

logout

Connection to linuxsrva closed.

**student@localpc ~ $**

1. Σαν root στο linuxsrvb, επεξεργαστείτε αρχείο **/etc/sysconfig/network-scripts/ifcfg-enp0s25**, προσθέτοντας μια επιπλέον διεύθυνση, την **10.0.1.2/24** και κατόπιν επαναφορτώστε τις νέες ρυθμίσεις.
   1. Συνδεθείτε στο linuxsrvb μέσω **ssh**, σαν χρήστης student.

**student@localpc ~ $** ssh student@linuxsrvb

[ … ]

**student@linuxsrvb ~ $**

* 1. Χρησιμοποιώντας την εντολή **sudo -i**, αλλάξτε στον χρήστη root.

**student@linuxsrvb ~ $** sudo -i

Password:

**root@linuxsrvb ~ #**

* 1. Τροποποιήστε το αρχείο **/etc/sysconfig/network-scripts/ifcfg-enp0s25**, προσθέτοντας τη δεύτερη διεύθυνση και το πρόθεμα δικτύου.

**root@linuxsrvb ~ #** echo "IPADDR2=10.0.1.2" >> \

> /etc/sysconfig/network-scripts/ifcfg-enp0s25

**root@linuxsrvb ~ #** echo "PREFIX2=24" >> /etc/sysconfig/network-scripts/ifcfg-enp0s25

* 1. Επαναφορτώστε τις αλλαγές ρυθμίσεων.

**root@linuxsrvb ~ #** nmcli con reload

* 1. Επανεκκινήστε τη σύνδεση με τις νέες ρυθμίσεις.

**root@linuxsrvb ~ #** nmcli con up "enp0s25"

Connection successfully activated (D-Bus active path: /org/freedesktop/

NetworkManager/ActiveConnection/3)

* 1. Επιβεβαιώστε ότι δόθηκε η νέα διεύθυνση.

**root@linuxsrvb ~ #** ip addr show enp0s25

3: enp0s25: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether f0:de:f1:5f:02:27 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.72/24 brd 192.168.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet 10.0.1.2/24 brd 10.0.1.255 scope global noprefixroute enp0s25

valid\_lft forever preferred\_lft forever

inet6 fe80::f2de:f1ff:ff5f:227/64 scope link proto kernel\_ll

valid\_lft forever preferred\_lft forever

1. Ελέγξτε τη συνδεσιμότητα χρησιμοποιώντας τις νέες διευθύνσεις δικτύου.
   1. Από τον linuxsrvb, κάντε ping στη νέα διεύθυνση του linuxsrva.

**root@linuxsrvb ~ #** ping -c3 10.0.1.1

PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.

64 bytes from 10.0.1.1: icmp\_seq=1 ttl=128 time=1.70 ms

64 bytes from 10.0.1.1: icmp\_seq=2 ttl=128 time=1.94 ms

64 bytes from 10.0.1.1: icmp\_seq=3 ttl=128 time=1.95 ms

--- 10.0.1.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2003ms

rtt min/avg/max/mdev = 1.702/1.863/1.950/0.114 ms

* 1. Αποσυνδεθείτε από τον linuxsrvb.

**root@linuxsrvb ~ #** exit

logout

**student@linuxsrvb ~ $** exit

logout

Connection to linuxsrvb closed.

**student@localpc ~ $**

* 1. Συνδεθείτε στο linuxsrva μέσω **ssh**, σαν χρήστης student.

**student@localpc ~ $** ssh student@linuxsrva

[ … ]

**student@linuxsrva ~ $**

* 1. Χρησιμοποιώντας την εντολή **sudo -i**, αλλάξτε στον χρήστη root.

**student@linuxsrva ~ $** sudo -i

Password:

**root@linuxsrva ~ #**

* 1. Από τον linuxsrva, κάντε ping στη νέα διεύθυνση του linuxsrvb.

**root@linuxsrva ~ #** ping -c3 10.0.1.2

PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.

64 bytes from 10.0.1.2: icmp\_seq=1 ttl=128 time=2.09 ms

64 bytes from 10.0.1.2: icmp\_seq=2 ttl=128 time=2.47 ms

64 bytes from 10.0.1.2: icmp\_seq=3 ttl=128 time=2.04 ms

--- 10.0.1.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 1987ms

rtt min/avg/max/mdev = 2.036/2.198/2.472/0.194 ms

* 1. Αποσυνδεθείτε από τον linuxsrva.

**root@linuxsrva ~ #** exit

logout

**student@linuxsrva ~ $** exit

logout

Connection to linuxsrva closed.

**student@localpc ~ $**

## ΡΥΘΜΙΖΟΝΤΑΣ ΟΝΟΜΑΤΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΝ ΕΠΙΛΥΣΗ ΟΝΟΜΑΤΩΝ

Σε αυτή την ενότητα αναλύεται πώς μπορεί να ρυθμιστεί το στατικό όνομα ενός εξυπηρετητή και η επίλυση του ονόματός του και να ελέγχθούν τα αποτελέσματα.

### ΑΛΛΑΖΟΝΤΑΣ ΤΟ ΟΝΟΜΑ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ

Η εντολή **hostname** εμφανίζει το πλήρες όνομα χώρου του υπολογιστή, ενώ μπορεί και να το αλλάξει προσωρινά.

**root@linuxsrv:~#** hostname

linuxsrv@some-domain.com

Το στατικό (σταθερό) όνομα μπορεί να οριστεί στο αρχείο **/etc/hostname**. Τα περιεχόμενα του αρχείου αυτού μπορούν να αλλαχτούν με την εντολή **hostnamectl**. Με την ίδια εντολή μπορεί να εμφανιστεί το πλήρες πιστοποιημένο όνομα του υπολογιστή.

**root@linuxsrv:~#** hostnamectl set-hostname linuxsrv@some-domain.com

**root@linuxsrv:~#** hostnamectl

Static hostname: linuxsrv.some-domain.com

Icon name: computer-desktop

Chassis: desktop

Machine ID: 94539adab0944f8ab23d8779879d0467

Boot ID: d2b26ac90bc726329545b3adae31132b

Operating System: Red Hat Enterprise Linux 8.0 beta (Ootpa)

Kernel: Linux 4.18.0-60.el8.x86\_64

Architecture: x86-64

**root@linuxsrv:~#** cat /etc/hostname

linuxsrv@some-domain.com

Να σημειώσουμε εδώ ότι σε εκδόσεις του Red Hat Enterprise Linux παλιότερες της έκδοσης 7, το όνομα του υολογιστή δε φυλάσσεται στο **/etc/hostname**, αλλά σε μια από τις μεταβλητές του αρχείου **/etc/sysconfig/network**.

### ΡΥΘΜΙΖΟΝΤΑΣ ΤΗΝ ΕΠΙΛΥΣΗ ΟΝΟΜΑΤΩΝ

O *stub resolver – επιλύτης στελέχους* χρησιμοποιείται για να μετατρέψει ονόματα υπολογιστών σε διευθύνσεις IP και το αντίστροφο. Το πού θα ψάξει καθορίζεται από τη ρύθμιση του αρχείου **/etc/nsswitch.conf**. Η προκαθορισμένη σειρά ορίζει να ελέγχονται πρώτα τα περιεχόμενα του αρχείου **/etc/hosts**.

**root@linuxsrv:~#** cat /etc/hosts

127.0.0.1 localhost

::1 localhost

10.1.83.53 srv1.some-domain.com

10.1.9.249 srv2.some-domain.com

Για να ελέγξουμε την επίλυση ονομάτων με χρήση του **/etc/hosts** χρησιμοποιούμε την εντολή **getent hosts *όνομα-υπολογιστή***.

Αν δε βρεθεί καταχώρηση στο αρχείο **/etc/hosts**, η προκαθορισμένη λειτουργία του stub resolver είναι να ψάξει το όνομα χρησιμοποιώντας έναν εξυπηρετητή DNS. Το πώς ακριβώς εκτελείται το ερώτημα ελέγχεται από το αρχείο **/etc/resolv.conf**.

Η καταχώρηση seach περιλαμβάνει μια λίστα από ονόματα χώρων (domain names) τα οποία δοκιμάζει με ένα σύντομο (short) όνομα υπολογιστή. Η καταχώρηση αυτή δεν πρέπει να βρίσκεται στο ίδιο αρχείο ταυτόχρονα με την καταχώρηση domain. Αν βρεθούν, τότε υπερισχύει η τελευταία καταχώρηση. Για περισσότερες λεπτομέρειες μπορείτε να δείτε το resolv.conf(5).

Η καταχώρηση nameserver δίνει τη διεύθυνση IP ενός εξυπηρετητή ονομάτων, που μπορεί να ερωτηθεί. Μπορούν να υπάρχουν μέχρι τρεις τέτοιες καταχωρήσεις, ώστε να προσφέρουν εναλλακτικές σε περίπτωση που κάποιος εξυπηρετητής είναι εκτός.

**root@linuxsrv ~#** cat /etc/resolv.conf

# Generated by NetworkManager

domain some-domain.com

nameserver 1.1.1.1

nameserver 1.0.0.1

Το NetworkManager ενημερώνει το αρχείο **/etc/resolv.conf** χρησιμοποιώντας τις ρυθμίσεις από τα αρχεία ρυθμίσεων συνδέσεων. Όπως έχει αναφερθεί, τα περιεχόμενά τους μπορούν να αλλαχτούν με την εντολή **nmcli**. Για την αλλαγή των ρυθμίσεων που αφορούν το DNS οι σχετικές εντολές είναι:

**root@linuxsrv ~#** nmcli con mod *ID* ipv4.dns *IP*

**root@linuxsrv ~#** nmcli con down *ID*

**root@linuxsrv ~#** nmcli con up *ID*

**root@linuxsrv ~#** cat /etc/sysconfig/network-scripts/ifcfg-*ID*

( … )

DNS1=1.1.1.1

( … )

Η προκαθορισμένη συμπεριφορά της **nmcli con mod ID ipv4.dns IP** είναι η αντικατάσταση των προηγούμενων ρυθμίσεων DNS με τη νέα λίστα διευθύνσεων της εντολής. Αν μπροστά από το όρισμα **ipv4.dns** υπάρχει ένα σύμβολο + ή -, τότε αυτή η καταχώρηση προστίθεται ή αφαιρείται, αντίστοιχα.

**root@linuxsrv ~#** nmcli con mod *ID* +ipv4.dns 8.8.8.8

Για να προσθέσουμε τον εξυπηρετητή DNS με διεύθυνση IPv6 2606:4700:4700::1111 στη λίστα που χρησιμοποιεί η σύνδεση enp0s25- static, δίνουμε την εντολή:

**root@linuxsrv ~#** nmcli con mod enp0s25-static +ipv6.dns 2606:4700:4700::1111

Όλες οι στατικές ρυθμίσεις DNS για IPv4 και IPv6 τελικά γίνονται καταχωρήσεις nameserver στο **/etc/resolv.conf**. Σε συστήματα που χρησιμοποιούν και τις δύο εκδόσεις του IP («dual stack»),πρέπει να υπάρχει τουλάχιστον ένας εξυπηρετητής DNS προσβάσιμος από το IPv4, ενώ ένας δεύτερος, προσβάσιμος από το IPv6 εξασφαλίζει συνεχή πρόσβαση σε εξυπηρετητές DNS σε περιπτώσεις που παρουσιάζει προβλήματα είτε το IPv4, είτε το IPv6.

Όταν ενεργοποιείται το DHCP, γράφει πάνω από τα περιεχόμενα του **/etc/resolv.conf** τη στιγμή που ενεργοποιούνται οι διεπαφές. Αν δεν είναι επιθυμητή αυτή η συμπεριφορά, αρκεί να θέσουμε τη ρύθμιση PEERDNS=no στα σχετικά αρχεία ρυθμίσεων, με την παρακάτω εντολή **nmcli**:

**root@linuxsrv ~#** nmcli con mod static-enp0s20 ipv4.ignore-auto-dns yes

#### Ελέγχοντας τη λειτουργία του DNS

Για να ελέγξουμε κατά πόσο η συνδεσή μας με εξυπηρετητές DNS είναι ενεργή, χρησιμοποιούμε την εντολή **host *όνομα-υπολογιστή***.

**root@linuxsrv ~#** host srv1.some-domain.com

srv1.some-domain.com has address 10.1.83.53

**agio@mylinux:~$** host 10.1.83.53

10.1.83.53.in-addr.arpa domain name srv1.some-domain.com.

**Προσοχή:** το DHCP αλλάζει τα περιεχόμενα του **/etc/resolv.conf** όταν ενεργοποιούνται διεπαφές. Αν αυτό δεν είναι επιθυμητό, πρέπει στις ρυθμίσεις της αντίστοιχης διεπαφής να ορίσουμε «PEERDNS=no». μπορεί να γίνει με την εντολή **nmcli**.

**root@linuxsrv ~#** nmcli con mod "enp0s25-static" ipv4.ignore-auto-dns yes

## ΚΑΘΟΔΗΓΟΥΜΕΝΗ ΑΣΚΗΣΗ ΡΥΘΜΙΣΗ ΟΝΟΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΠΙΛΥΣΗΣ ΟΝΟΜΑΤΩΝ

Σε αυτή την άσκηση θα επεξεργαστείτε χειροκίνητα (χωρίς τη χρήση ειδικών εργαλείων ρύθμισης) το στατικό όνομα του υπολογιστή (static host name), το αρχείο **/etc/hosts** και τον επιλύτη ονομάτων DNS**.**

1. Αποτελέσματα:

Θα πρέπει να μπορείτε να ορίζετε το επιθυμητό όνομα υπολογιστή και να ορίζετε τις ρυθμίσεις επίλυσης ονομάτων.

1. Συνδεθείτε στον linuxsrva σαν student. Αν έχει ρυθμιστεί κλειδί SSH για την είσοδο, δεν θα χρειαστεί να δώσετε συνθηματικό.

**student@localpc ~ $** ssh student@linuxsrva

**student@linuxsrva ~ $**

1. Εμφανίστε τις τρέχουσες ρυθμίσεις για το όνομα του υπολογιστή.
   1. Εμφανίστε το τρέχον όνομα του υπολογιστή.

**student@linuxsrva ~ $** hostname

linuxsrv.some-domain.com

* 1. Εμφανίστε την κατάσταση του ονόματος του υπολογιστή.

**student@linuxsrva ~ $** hostnamectl status

Static hostname: **linuxsrva**

Icon name: computer-vm

Chassis: vm

Machine ID: 45f0b749fc4946faa64da72848ce09ef

Boot ID: 3275b255d17649d3b3031c2d12de630a

Virtualization: microsoft

Operating System: AlmaLinux 9.5 (Teal Serval)

CPE OS Name: cpe:/o:almalinux:almalinux:9::baseos

Kernel: Linux 5.14.0-503.33.1.el9\_5.x86\_64

Architecture: x86-64

1. Ορίστε ένα στατικό όνομα υπολογιστή, ίδιο με το τρέχον προσωρινό όνομα.
   1. Αλλάξτε το όνομα υπολογιστή και το αρχείο ρυθμίσεων ονόματος υπολογιστή.

**student@linuxsrva ~ $** sudo hostnamectl set-hostname linuxsrva.some-domain.com

[sudo] password for student:

**student@linuxsrva ~ $**

* 1. Εμφανίστε το αρχείο ρυθμίσεων που παρέχει το όνομα υπολογιστή κατά την εκκίνηση του δικτύου.

**student@linuxsrva ~ $** cat /etc/hostname

linuxsrv.some-domain.com

* 1. Εμφανίστε την κατάσταση του ονόματος υπολογιστή.

**student@linuxsrva ~ $** hostnamectl status

Static hostname: **linuxsrva**

Icon name: computer-vm

Chassis: vm

Machine ID: 45f0b749fc4946faa64da72848ce09ef

Boot ID: 3275b255d17649d3b3031c2d12de630a

Virtualization: microsoft

Operating System: AlmaLinux 9.5 (Teal Serval)

CPE OS Name: cpe:/o:almalinux:almalinux:9::baseos

Kernel: Linux 5.14.0-503.33.1.el9\_5.x86\_64

Architecture: x86-64

1. Αλλάξτε προσωρινά το όνομα του υπολογιστή.
   1. Αλλάξτε το όνομα.

**student@linuxsrva ~ $** sudo hostname tempname

tempname

* 1. Εμφανίστε το τρέχον όνομα υπολογιστή.

**student@linuxsrva ~ $** hostname

tempname

* 1. Εμφανίστε το αρχείο ρυθμίσεων που δίνει το όνομα του υπολογιστή κατά την εκκίνηση του δικτύου.

**student@linuxsrva ~ $** cat /etc/hostname

linuxsrv.some-domain.com

* 1. Επανεκκινήστε το μηχάνημα.

**student@linuxsrva ~ $** sudo reboot

Connection to linuxsrva closed by remote host.

Connection to linuxsrva closed.

**student@localpc ~ $**

* 1. Ξανασυνδεθείτε στον linuxsrva σαν student.

**student@localpc ~ $** ssh student@**linuxsrva**

**student@linuxsrva ~ $**

* 1. Εμφανίστε το τρέχον όνομα του υπολογιστή.

**student@linuxsrva ~ $** hostname

linuxsrv.some-domain.com

1. Προσθέστε ένα παρατσούκλι για τον τοπικό υπολογιστή.
   1. Βρείτε τη διεύθυνση IP του localpc.some-domain.com.

**student@linuxsrva ~ $** host localpc.some-domain.com

localpc.some-domain.com has address 192.168.1.251

* 1. Τροποποιήστε το **/etc/hosts**, έτσι ώστε το όνομα **classsrv** να έχει διεύθυνση IP 192.168.1.254 και να μπορεί να χρησιμοποιηθεί για την επικοινωνία με το **classsrv.some-domain.com.**

**student@linuxsrva ~ $** sudo vim /etc/hosts

**student@linuxsrva ~ $** cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.1.254 classsrv.some-domain.com classsrv

[ … ]

* 1. Αναζητήστε τη διεύθυνση IP του classsrv.

**student@linuxsrva ~ $** host classsrv

Host classsrv not found: 3(NXDOMAIN)

**student@linuxsrva ~ $** getent hosts classsrv

192.168.1.254 classsrv.some-domain.com classsrv

* 1. Κάντε ping στο classsrv.

agio@AxisCity ~ $ ping -c3 classsrv

PING classsrv (192.168.1.254) 56(84) bytes of data.

64 bytes from classsrv (192.168.1.254): icmp\_seq=1 ttl=64 time=0.250 ms

64 bytes from classsrv (192.168.1.254): icmp\_seq=2 ttl=64 time=0.321 ms

64 bytes from classsrv (192.168.1.254): icmp\_seq=3 ttl=64 time=0.284 ms

--- B5 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2039ms

rtt min/avg/max/mdev = 0.250/0.285/0.321/0.029 ms

* 1. Βγείτε από το linuxsrva.

# Εγκατάσταση και ενημέρωση πακέτων λογισμικού – Ρύθμιση της πρόσβασης στο σύστημα των αρχείων

## Εγκατάσταση και ενημέρωση πακέτων λογισμικού

Η εγκατάσταση μιας διανομής Linux περιλαμβάνει όχι μόνο τον πυρήνα (**Kernel**), αλλά και πλήθος άλλων λογισμικών που αναλαμβάνουν διάφορες περιφερειακές εργασίες και λειτουργίες. Ωστόσο, συχνά ο χρήστης ενδέχεται να χρειαστεί να εγκαταστήσει νέα λογισμικά στο σύστημα. Η διαδικασία εγκατάστασης λογισμικών στο Linux είναι σε μεγάλο βαθμό αυτοματοποιημένη και απλοποιημένη μέσω εργαλείων διαχείρισης πακέτων.

### Βασικές έννοιες των πακέτων λογισμικού

Στις περισσότερες διανομές Linux, τα λογισμικά διανέμονται με τη μορφή **πακέτων λογισμικού** (software packages). Ένα πακέτο λογισμικού είναι ένα συμπιεσμένο αρχείο που περιλαμβάνει όλα τα αρχεία που απαιτούνται για την εγκατάσταση και εκτέλεση ενός προγράμματος στο σύστημα. Αυτό μπορεί να περιλαμβάνει εκτελέσιμα αρχεία, αρχεία ρυθμίσεων, σενάρια (scripts), καθώς και άλλα μεταδεδομένα που είναι απαραίτητα για την ορθή εγκατάσταση και λειτουργία του λογισμικού.

Τα **μεταδεδομένα** ενός πακέτου περιλαμβάνουν σημαντικές πληροφορίες, όπως:

1. Ο κατάλογος στον οποίο πρέπει να εγκατασταθεί το λογισμικό.
2. Η έκδοση του λογισμικού και άλλα χαρακτηριστικά.
3. Η ανάγκη για εγκατάσταση άλλων προγραμμάτων ή βιβλιοθηκών πριν από την εγκατάσταση του πακέτου, γνωστό ως **εξάρτηση λογισμικού** (software dependency).

Τα πακέτα λογισμικού είναι οργανωμένα και αποθηκεύονται σε **αποθετήρια λογισμικού** (software repositories). Τα αποθετήρια αυτά είναι μεγάλες βάσεις δεδομένων που περιέχουν πλήθος διαθέσιμων πακέτων που μπορούν να κατεβούν και να εγκατασταθούν στο σύστημα. Πολλές διανομές Linux, όπως το Rocky Linux, διαθέτουν επίσημα αποθετήρια που περιλαμβάνουν πακέτα που έχουν δοκιμαστεί και πιστοποιηθεί για την ομαλή λειτουργία τους στη συγκεκριμένη διανομή.

Οι διαφορετικές διανομές Linux υιοθετούν ξεχωριστά πρότυπα για τη δομή των πακέτων λογισμικού, κάτι που οδηγεί σε δύο κύριες «οικογένειες»:

Α) Red Hat-based διανομές: Χρησιμοποιούν πακέτα με κατάληξη .rpm (Red Hat Package Manager).

Β) Debian-based διανομές: Χρησιμοποιούν πακέτα με κατάληξη .deb, που διαχειρίζονται μέσω του συστήματος APT (Advanced Package Tool).

Αυτή η διαφοροποίηση καθορίζει και τα εργαλεία διαχείρισης πακέτων, όπως το dnf για RPM-based διανομές και το apt για Debian-based διανομές.

**Παράδειγμα Ονοματοδοσίας Πακέτου Λογισμικού**

Η ονομασία ενός πακέτου στο Linux περιλαμβάνει χρήσιμες πληροφορίες που βοηθούν τον χρήστη να αναγνωρίσει τη φύση του. Ας δούμε ένα παράδειγμα:

nano-5.6.1-6.el9.x86\_64.rpm

* nano: Το όνομα του λογισμικού, στην προκειμένη περίπτωση ο επεξεργαστής κειμένου Nano.
* 5.6.1: Η έκδοση (version) του λογισμικού, όπως ορίστηκε από τον δημιουργό του.
* 6.el9: Ο αριθμός διανομής (release), που προσδιορίζεται από τον ιδιοκτήτη του αποθετηρίου. Εδώ, el9 υποδηλώνει ότι το πακέτο προορίζεται για το Red Hat Enterprise Linux 9 και τις σχετικές διανομές.
* x86\_64: Η αρχιτεκτονική επεξεργαστή για την οποία έχει συνταχθεί (compiled) το πακέτο. Εδώ υποδεικνύει ότι προορίζεται για 64-bit συστήματα.

### Εργαλεία διαχείρισης πακέτων λογισμικού

Η διαχείριση πακέτων λογισμικού σε ένα σύστημα Linux πραγματοποιείται μέσω δύο ειδών εργαλείων:

* Εργαλείο χαμηλού επιπέδου: Αναλαμβάνει τη βασική λειτουργία εγκατάστασης των πακέτων, όπως το άνοιγμα των αρχείων, την εκτέλεση των scripts εγκατάστασης και τη μεταφορά των αρχείων στο σύστημα.
* Εργαλείο υψηλού επιπέδου: Εστιάζει στη διαχείριση των εξαρτήσεων μεταξύ πακέτων και στη διαχείριση των αποθετηρίων λογισμικού.

Στο Rocky Linux (και γενικότερα στις διανομές που βασίζονται στο Red Hat), το εργαλείο χαμηλού επιπέδου είναι το **rpm** (Red Hat Package Manager). Αντίστοιχα, το εργαλείο υψηλού επιπέδου, που ήταν παλαιότερα το **yum**, έχει πλέον αντικατασταθεί από το **DNF**, το οποίο προσφέρει βελτιωμένη απόδοση και σύγχρονες δυνατότητες.

Προβολή πληροφοριών πακέτου λογισμικού

Μπορούμε να εξετάσουμε πληροφορίες για ένα πακέτο λογισμικού που βρίσκεται ήδη στον υπολογιστή μας χρησιμοποιώντας την εντολή rpm -qpi. Ακολουθεί ένα παράδειγμα:

[user@host ~]$ rpm -qpi nano-5.6.1-6.el9.x86\_64.rpm

Name : nano

Version : 5.6.1

Release : 6.el9

Architecture: x86\_64

….

URL : https://www.nano-editor.org

Summary : A small text editor

Description :

GNU nano is a small and friendly text editor.

[user@host ~]$

* **Name**: Το όνομα του πακέτου, στην περίπτωση αυτή ο επεξεργαστής κειμένου Nano.
* **Version και Release**: Η έκδοση του λογισμικού και η κυκλοφορία της διανομής του.
* **Architecture**: Η αρχιτεκτονική για την οποία έχει συνταχθεί (x86\_64 για 64-bit συστήματα).
* **URL και Summary**: Ο ιστότοπος του λογισμικού και μια συνοπτική περιγραφή του.
* **Description**: Μια εκτενέστερη περιγραφή του λογισμικού.

Για να δούμε τα αρχεία που περιλαμβάνει ένα πακέτο και πού θα εγκατασταθούν, χρησιμοποιούμε την εντολή rpm -qpl:

[user@host ~]$ rpm -qpl nano-5.6.1-6.el9.x86\_64.rpm

/etc/nanorc

/usr/bin/nano

/usr/bin/rnano

/usr/lib/.build-id

/usr/lib/.build-id/bf

…

Είναι δυνατόν να αναζητήσουμε τα περιεχόμενα ενός πακέτου απευθείας από το αποθετήριο, χωρίς να χρειάζεται να το κατεβάσουμε πρώτα. Αυτό γίνεται μέσω της εντολής dnf repoquery -l:

[user@host ~]$ dnf repoquery -l nano

Rocky Linux 9 - BaseOS 1.7 MB/s | 2.3 MB 00:01

Rocky Linux 9 - AppStream 2.3 MB/s | 8.3 MB 00:03

Rocky Linux 9 - Extras 40 kB/s | 15 kB 00:00

/etc/nanorc

/usr/bin/nano

/usr/bin/rnano

/usr/lib/.build-id

/usr/lib/.build-id/bf

…

Ο έλεγχος ακεραιότητας ενός πακέτου λογισμικού είναι σημαντικός για να επιβεβαιωθεί ότι το πακέτο δεν έχει υποστεί αλλοίωση κατά τη μεταφορά ή την αποθήκευσή του. Τα πακέτα RPM περιέχουν ψηφιακές υπογραφές και **checksums**, τα οποία μπορούμε να επαληθεύσουμε με την εντολή rpm -K (ή --checksig).

Παράδειγμα ελέγχου:

[user@host ~]$ rpm -K popt-1.18-8.el9.x86\_64.rpm

popt-1.18-8.el9.x86\_64.rpm: digests signatures OK

Η ένδειξη OK επιβεβαιώνει ότι το πακέτο είναι έγκυρο και δεν έχει αλλοιωθεί.

### Εγκατάσταση / απεγκατάσταση / ενημέρωση πακέτων λογισμικού

**Εγκατάσταση πακέτου λογισμικού**

Η εγκατάσταση ενός πακέτου λογισμικού στο Linux γίνεται με τη χρήση της εντολής dnf install. Η γενική σύνταξη της εντολής είναι:

dnf install [επιλογές] <όνομα πακέτου>

Για παράδειγμα, ακολουθεί η εγκατάσταση του λογισμικού iotop:

[user@host ~]$ sudo dnf install iotop

Last metadata expiration check: 1:05:48 ago on Tue 26 Nov 2024 07:22:30 AM EET.

Dependencies resolved.

==============================================================================

Package Architecture Version Repository Size

==============================================================================

Installing:

iotop noarch 0.6-30.el9 baseos 62 k

Transaction Summary

==============================================================================

Install 1 Package

Total download size: 62 k

Installed size: 173 k

Is this ok [y/N]: y

Downloading Packages:

… (η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου)

Ανάλυση της Διαδικασίας:

* **Έλεγχος εξαρτήσεων**: Το σύστημα εντοπίζει τα πακέτα που απαιτούνται για τη σωστή λειτουργία του λογισμικού που πρόκειται να εγκατασταθεί.
* **Εμφάνιση πληροφοριών**: Εμφανίζονται λεπτομέρειες για το πακέτο, όπως η έκδοση, το μέγεθος λήψης και ο χώρος που θα καταλάβει στο δίσκο μετά την εγκατάσταση.
* **Επιβεβαίωση χρήστη**: Το σύστημα ζητά την επιβεβαίωση του χρήστη (προεπιλεγμένο: No). Πληκτρολογώντας y, ξεκινά η εγκατάσταση.

**Εγκατάσταση πολλαπλών πακέτων με μεταχαρακτήρες**: Η εγκατάσταση πολλών πακέτων μπορεί να γίνει χρησιμοποιώντας μεταχαρακτήρες. Για παράδειγμα, για την εγκατάσταση όλων των πακέτων που το όνομά τους ξεκινά με τη λέξη gimp, χρησιμοποιούμε:

[user@host ~]$ sudo dnf install gimp\\*

Σημείωση: Ο χαρακτήρας \ χρησιμοποιείται για να αποτρέψει την αντικατάσταση των μεταχαρακτήρων με ονόματα αρχείων από το shell (file globbing).

**Λήψη πακέτου χωρίς εγκατάσταση**: Εάν θέλουμε μόνο να κατεβάσουμε το συμπιεσμένο αρχείο ενός πακέτου (χωρίς να το εγκαταστήσουμε), χρησιμοποιούμε την εντολή dnf download. Για παράδειγμα:

[user@host ~]$ dnf download mutt

Last metadata expiration check: 1:34:38 ago on Tue 26 Nov 2024 08:11:56 AM EET.

mutt-2.2.6-2.el9.x86\_64.rpm 3.0 MB/s | 1.9 MB 00:00

**Απεγκατάσταση πακέτου λογισμικού**

Η απεγκατάσταση ενός πακέτου γίνεται με την εντολή dnf remove, η οποία αφαιρεί το πακέτο και τυχόν εξαρτήσεις που δεν χρησιμοποιούνται πλέον από άλλα προγράμματα. Η σύνταξη είναι:

dnf install [επιλογές] <όνομα πακέτου>

Για παράδειγμα, για την αφαίρεση του πακέτου iotop:

[user@host ~]$ sudo dnf remove iotop

Το σύστημα θα εμφανίσει μια λίστα με τα αρχεία που θα αφαιρεθούν και θα ζητήσει την επιβεβαίωση του χρήστη.

**Ενημέρωση πακέτου λογισμικού**

Η ενημέρωση ενός πακέτου περιλαμβάνει δύο βήματα:

**Έλεγχος για νέες εκδόσεις**: Με την εντολή dnf check-update, μπορούμε να δούμε αν υπάρχει διαθέσιμη ενημέρωση για συγκεκριμένο λογισμικό ή για όλα τα πακέτα του συστήματος:

Παράδειγμα ελέγχου για το πακέτο lvm2:

[user@host ~]$ dnf check-update lvm2

Last metadata expiration check: 2:02:59 ago on Tue 26 Nov 2024 08:11:56 AM EET.

lvm2.x86\_64 9:2.03.24-2.el9 baseos

**Ενημέρωση του λογισμικού**: Αν υπάρχει διαθέσιμη νέα έκδοση, μπορούμε να την εγκαταστήσουμε με την εντολή dnf upgrade:

Παράδειγμα αναβάθμισης του lvm2:

[user@host ~]$ sudo dnf upgrade lvm2

[sudo] password for user:

Last metadata expiration check: 2:52:37 ago on Tue 26 Nov 2024 07:22:30 AM EET.

Dependencies resolved.

... (έξοδος έχει περικοπεί για λόγους οικονομίας χώρου)

Κατά την αναβάθμιση, το σύστημα αναλύει τις εξαρτήσεις, εμφανίζει τις αλλαγές και ζητά την επιβεβαίωση του χρήστη πριν προχωρήσει.

### Αναζήτηση πακέτων λογισμικού – Προβολή λιστών λογισμικού

Η διαχείριση πακέτων στο Linux μέσω της dnf περιλαμβάνει ισχυρές δυνατότητες αναζήτησης, τόσο για πακέτα που βρίσκονται στην τοπική βάση δεδομένων του συστήματος όσο και για αυτά που είναι διαθέσιμα στα αποθετήρια (repositories).

**Αναζήτηση με λέξεις-κλειδιά**

Όταν δεν γνωρίζουμε το ακριβές όνομα ενός πακέτου λογισμικού, μπορούμε να χρησιμοποιήσουμε την εντολή dnf search για αναζήτηση με βάση λέξεις-κλειδιά. Η εντολή αυτή επιστρέφει πακέτα που περιέχουν τις λέξεις-κλειδιά είτε στο όνομα είτε στην περιγραφή τους.

Παράδειγμα: Αναζήτηση πακέτων που σχετίζονται με τη λέξη shell:

[user@host ~]$ dnf search shell

Η έξοδος θα περιλαμβάνει μια λίστα πακέτων που περιέχουν τη λέξη shell στο όνομα ή στη σύντομη περιγραφή τους.

**Εύρεση πακέτου λογισμικού που περιέχει μια εντολή ή ένα αρχείο**

Σε περιπτώσεις όπου γνωρίζουμε το όνομα μιας εντολής (ή ενός αρχείου), αλλά δεν γνωρίζουμε ποιο πακέτο περιέχει την εντολή, μπορούμε να χρησιμοποιήσουμε την εντολή dnf provides.

Παράδειγμα: Εντοπισμός του πακέτου που περιέχει την εντολή sar:

[user@host ~]$ dnf provides /usr/bin/sar

Last metadata expiration check: 0:14:08 ago on Wed 11 Dec 2024 07:39:17 AM EET.

sysstat-12.5.4-9.el9.x86\_64 : Collection of performance monitoring tools for

: Linux

Repo : appstream

Matched from:

Filename : /usr/bin/sar

Η εντολή μας δείχνει ότι η εκτελέσιμη εντολή sar βρίσκεται στο πακέτο sysstat, το οποίο μπορεί να εγκατασταθεί από το αποθετήριο appstream.

**Δημιουργία λιστών πακέτων**

Η εντολή dnf list μας επιτρέπει να δημιουργούμε λίστες πακέτων λογισμικού που είναι είτε εγκατεστημένα στο σύστημά μας είτε διαθέσιμα για εγκατάσταση στα αποθετήρια. Η γενική σύνταξη της εντολής είναι:

dnf list [επιλογή]

Συχνά χρησιμοποιούμενες επιλογές

* **installed**: Εμφανίζει όλα τα πακέτα που είναι εγκατεστημένα στο σύστημα.
* **available**: Εμφανίζει τα πακέτα που είναι διαθέσιμα προς εγκατάσταση στα αποθετήρια.
* **updates**: Δείχνει ποια πακέτα έχουν διαθέσιμες ενημερώσεις.
* **extras**: Εμφανίζει πακέτα που είναι εγκατεστημένα στο σύστημα αλλά δεν υπάρχουν σε κανένα αποθετήριο.
* **recent**: Εμφανίζει πακέτα που προστέθηκαν πρόσφατα στα αποθετήρια.

Παραδείγματα χρήσης

Για να δούμε ποια πακέτα λογισμικού είναι εγκατεστημένα στο σύστημα μπορούμε να εκτελέσουμε:

[user@host ~]$ dnf list installed

Για να δούμε ποια πακέτα είναι διαθέσιμα προς εγκατάσταση στα αποθετήρια εκτελούμε:

[user@host ~]$ dnf list available

Για να δούμε ποια πακέτα χρειάζονται ενημέρωση:

[user@host ~]$ dnf list updates

Η εντολή dnf list μπορεί να συνδυαστεί με εντολές όπως grep για να φιλτράρουμε αποτελέσματα.

Παράδειγμα: Αναζήτηση για την έκδοση του προγράμματος firefox που είναι εγκατεστημένη:

[user@host ~]$ dnf list installed | grep firefox

firefox.x86\_64 128.4.0-1.el9\_4 @appstream

[user@host ~]$

### Ομάδες (group) πακέτων λογισμικών

Εκτός από την εγκατάσταση μεμονωμένων πακέτων λογισμικού, το Linux παρέχει τη δυνατότητα διαχείρισης ομάδων πακέτων λογισμικού. Οι ομάδες αυτές περιέχουν πολλά πακέτα που συνεργάζονται για την εκπλήρωση ενός συγκεκριμένου σκοπού, όπως η εγκατάσταση ενός γραφικού περιβάλλοντος ή ενός συνόλου εργαλείων ασφαλείας.

Χαρακτηριστικά παραδείγματα τέτοιων ομάδων είναι τα γραφικά περιβάλλοντα Xfce, KDE, και Mate, καθώς και σύνολα εργαλείων όπως τα Development Tools ή τα Security Tools. Η διαχείριση αυτών των ομάδων γίνεται μέσω της εντολής dnf group, η οποία υποστηρίζει παρόμοιες λειτουργίες με αυτές για μεμονωμένα πακέτα.

Για να δούμε τις διαθέσιμες ομάδες λογισμικού που είναι είτε εγκατεστημένες στο σύστημα είτε διαθέσιμες στα αποθετήρια, χρησιμοποιούμε την εντολή:

[user@host ~]$ dnf group list

**Εγκατάσταση Ομάδων Λογισμικού**

Για την εγκατάσταση μιας ομάδας λογισμικών, χρησιμοποιούμε την εντολή dnf group install ακολουθούμενη από το όνομα της ομάδας. Το όνομα της ομάδας πρέπει να είναι περικλεισμένο σε διπλά εισαγωγικά αν περιέχει κενά.

Παράδειγμα: Εγκατάσταση της ομάδας λογισμικών System Tools:

[user@host ~]$ sudo dnf group install "System Tools"

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε την dnf install προσθέτοντας το σύμβολο @ πριν από το όνομα της ομάδας. Ο χαρακτήρας @ πριν από κάποιο όνομα καθορίζει στην dnf install ότι ακολουθεί το όνομα μιας ομάδας πακέτων λογισμικού.

Οι λειτουργίες διαχείρισης ομάδων πακέτων λογισμικού πραγματοποιούνται με την προσθήκη της λέξης group στις αντίστοιχες εντολές που χρησιμοποιούνται για τα μεμονωμένα πακέτα. Παρακάτω παρουσιάζεται ένας πίνακας με τις βασικές εντολές για τη διαχείριση ομάδων πακέτων λογισμικού:

|  |  |
| --- | --- |
| **Εντολή** | **Περιγραφή** |
| dnf group list [hidden] | Εμφανίζει τις διαθέσιμες και εγκατεστημένες ομάδες λογισμικού. |
| dnf group info "Ομάδα" | Παρέχει αναλυτικές πληροφορίες για μια συγκεκριμένη ομάδα λογισμικού. |
| sudo dnf group install "Ομάδα" | Εγκαθιστά μια ομάδα λογισμικού στο σύστημα. |
| sudo dnf group remove "Ομάδα" | Αφαιρεί μια ομάδα λογισμικού μαζί με τα σχετικά πακέτα της. |
| sudo dnf group upgrade "Ομάδα" | Ενημερώνει τα πακέτα μιας εγκατεστημένης ομάδας λογισμικού. |

### Ρυθμίσεις του DNF: Διαχείριση αποθετηρίων και παραμετροποίηση

Το **DNF** είναι ένας ισχυρός διαχειριστής πακέτων στο Linux, και η λειτουργία του μπορεί να προσαρμοστεί ώστε να ανταποκρίνεται καλύτερα στις ανάγκες του χρήστη. Οι βασικές ρυθμίσεις του περιλαμβάνουν τη διαχείριση αποθετηρίων (repositories), παραμετροποιήσεις μέσω αρχείων ρυθμίσεων, και τη διαχείριση προσωρινών δεδομένων (cache).

**Διαχείριση αποθετηρίων**

Τα αποθετήρια αποτελούν τις πηγές από τις οποίες το DNF ανακτά και εγκαθιστά πακέτα λογισμικού. Από προεπιλογή, το DNF χρησιμοποιεί τα επίσημα αποθετήρια της διανομής Linux που είναι εγκατεστημένη στο σύστημά μας. Μπορούμε να δούμε τα ενεργά αποθετήρια με την εντολή:

[user@host ~]$ dnf repolist

repo id repo name

appstream Rocky Linux 9 - AppStream

baseos Rocky Linux 9 - BaseOS

extras Rocky Linux 9 - Extras

Στην παραπάνω έξοδο παρατηρούμε ότι το κάθε αποθετήριο εκτός από το όνομα του έχει και ένα μικρό αναγνωριστικό (repo id) που το ξεχωρίζει από τα υπόλοιπα.

Η κατάσταση ενός αποθετηρίου μπορεί να τροποποιηθεί από ενεργό σε ανενεργό και αντίστροφα με τη χρήση των παρακάτω εντολών:

Ενεργοποίηση αποθετηρίου:

[user@host ~]$ sudo dnf config-manager --set-enabled plus

Απενεργοποίηση αποθετηρίου:

[user@host ~]$ sudo dnf config-manager --set-disabled plus

Ένα από τα πιο δημοφιλή πρόσθετα αποθετήρια για διανομές βασισμένες στο RHEL είναι το EPEL (Extra Packages for Enterprise Linux). Παρέχει μεγάλο αριθμό πακέτων λογισμικού που δεν περιλαμβάνονται στα επίσημα αποθετήρια. Η εγκατάστασή του γίνεται με:

sudo dnf install epel-release

**Αρχεία ρυθμίσεων DNF**

Οι βασικές ρυθμίσεις του DNF βρίσκονται στο αρχείο /etc/dnf/dnf.conf. Το αρχείο περιέχει:

* Ένα γενικό τμήμα [main], όπου ορίζονται οι ρυθμίσεις που ισχύουν για όλα τα αποθετήρια.
* Ξεχωριστά τμήματα [repository], τα οποία περιλαμβάνουν εξειδικευμένες ρυθμίσεις για κάθε αποθετήριο.

Συνίσταται η χρήση ξεχωριστών αρχείων .repo για κάθε αποθετήριο, τα οποία τοποθετούνται στον κατάλογο /etc/yum.repos.d/.

Μερικές από τις βασικές παραμέτρους που μπορούμε να ορίσουμε στο [main]:

* **debuglevel**: Καθορίζει το επίπεδο λεπτομέρειας στα μηνύματα debugging (0-10) (προεπιλογή: 2).
* **exclude**: Ορίζει πακέτα που αποκλείονται από την εγκατάσταση ή ενημέρωση (παράδειγμα exclude=python\*).
* **gpgcheck**: Ενεργοποιεί (1) ή απενεργοποιεί (0) τον έλεγχο ακεραιότητας μέσω υπογραφών.

Μερικές από τις βασικές ρυθμίσεις που ορίζει ένα [repository]:

* **name**: Περιγραφή του αποθετηρίου. Παράδειγμα: name=Rocky Linux 9 – BaseOS
* **baseurl**: Η διεύθυνση URL του αποθετηρίου.  
  Παράδειγμα: baseurl=http://mirror.centos.org/centos-9/9/BaseOS/x86\_64/os/
* **enabled**: Ενεργοποίηση (1) ή απενεργοποίηση (0) του αποθετηρίου.

Διαχείριση cache

Κατά τη λειτουργία του, το DNF δημιουργεί προσωρινά αρχεία (cache), όπως μεταδεδομένα αποθετηρίων και κατεβασμένα πακέτα. Η διαγραφή αυτών των δεδομένων γίνεται με την εντολή:

[user@host ~]$ dnf clean all

Για να διαγράψετε συγκεκριμένα είδη δεδομένων, χρησιμοποιήστε τις επιλογές:

* **all**: Διαγράφει όλα τα προσωρινά δεδομένα.
* **metadata**: Διαγράφει τα μεταδεδομένα αποθετηρίων.
* **packages**: Διαγράφει τα προσωρινά αποθηκευμένα πακέτα.

## Άσκηση: Εγκατάσταση και ενημέρωση πακέτων λογισμικού

### Εργασία 1: Εγκατάσταση / απεγκατάσταση / ενημέρωση πακέτων λογισμικού

Σκοπός της πρώτης εργασίας είναι η εξάσκηση σε σενάρια που σχετίζονται με την εγκατάσταση, ενημέρωση και απεγκατάσταση πακέτων λογισμικού σε ένα σύστημα Linux, χρησιμοποιώντας το εργαλείο διαχείρισης πακέτων DNF.

**Σενάριο 1: Εγκατάσταση ενός πακέτου λογισμικού**

Στόχος: Να εγκαταστήσουμε το πακέτο λογισμικού delve στο σύστημά μας, γνωρίζοντας το όνομά του. Θα εξετάσουμε επίσης πώς να εκτελέσουμε την εντολή σε μη-διαδραστική λειτουργία (non-interactive mode), παρακάμπτοντας τις ερωτήσεις επιβεβαίωσης. Αυτό είναι ιδιαίτερα χρήσιμο όταν θέλουμε να αυτοματοποιήσουμε τη διαδικασία εγκατάστασης μέσω scripts.

Η εγκατάσταση γίνεται με την εντολή dnf install. Εάν θέλουμε να παρακάμψουμε τις ερωτήσεις επιβεβαίωσης, χρησιμοποιούμε την επιλογή -y.

[user@host ~]$ sudo dnf install -y delve

Last metadata expiration check: 1:54:11 ago on Wed 11 Dec 2024 09:59:39 AM EET.

Dependencies resolved.

==============================================================================

Package Architecture Version Repository Size

==============================================================================

Installing:

delve x86\_64 1.22.1-1.el9 appstream 4.9 M

[η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου]

**Σενάριο 2: Ενημέρωση ενός πακέτου λογισμικού**

Στόχος: Να ελέγξουμε εάν υπάρχει διαθέσιμη νεότερη έκδοση ενός πακέτου λογισμικού που χρησιμοποιούμε (π.χ., firefox). Εάν υπάρχει, να προχωρήσουμε στην εγκατάστασή του.

Ελέγχουμε για διαθέσιμες ενημερώσεις με την εντολή dnf check-update:

[user@host ~]$ dnf check-update firefox

Last metadata expiration check: 4:17:08 ago on Wed 11 Dec 2024 07:39:17 AM EET.

firefox.x86\_64 128.5.1-1.el9\_5 appstream

Από την έξοδο της εντολής βλέπουμε ότι υπάρχει νεότερη έκδοση. Εγκαθιστούμε τη νέα έκδοση χρησιμοποιώντας την εντολή dnf upgrade. Η εντολή απαιτεί δικαιώματα διαχειριστή:

[user@host ~]$ sudo dnf upgrade firefox

[sudo] password for user:

Last metadata expiration check: 1:59:52 ago on Wed 11 Dec 2024 09:59:39 AM EET.

Dependencies resolved.

==============================================================================

Package Architecture Version Repository Size

==============================================================================

Upgrading:

firefox x86\_64 128.5.1-1.el9\_5 appstream 123 M

[η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου]

**Σενάριο 3: Απεγκατάσταση ενός πακέτου λογισμικού**

Στόχος: Να απεγκαταστήσουμε το πακέτο delve, το οποίο είναι ήδη εγκατεστημένο στο σύστημα μας.

Ελέγχουμε την ύπαρξη του εκτελέσιμου dlv, που σχετίζεται με το πακέτο delve, χρησιμοποιώντας την εντολή which:

[user@host ~]$ which dlv

/usr/bin/dlv

Αφαιρούμε το πακέτο delve χρησιμοποιώντας την εντολή dnf remove. Για να παρακάμψουμε την επιβεβαίωση, χρησιμοποιούμε την επιλογή -y:

[user@host ~]$ sudo dnf remove -y delve

[sudo] password for user:

Dependencies resolved.

==============================================================================

Package Architecture Version Repository Size

==============================================================================

Removing:

delve x86\_64 1.22.1-1.el9 @appstream 16 M

[η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου]

Επαληθεύουμε ότι το πακέτο έχει αφαιρεθεί επιτυχώς:

[user@host ~]$ which dlv

/usr/bin/which: no dlv in (/home/user/.local/bin:/home/user/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin)

### Εργασία 2: Αναζήτηση και δημιουργία λιστών με πακέτα λογισμικού

Σκοπός αυτής της ενότητας είναι η εξάσκηση σε σενάρια που αφορούν την αναζήτηση και παρουσίαση πακέτων λογισμικού στο Linux. Θα εξετάσουμε πώς να χρησιμοποιούμε τις εντολές dnf search και dnf list για να αποκτούμε πληροφορίες σχετικά με τα πακέτα και τη σχέση τους με το σύστημα.

**Σενάριο 1: Αναζήτηση πακέτων λογισμικού με βάση την περιγραφή**

Στόχος: Να βρούμε διαθέσιμους επεξεργαστές κειμένου στα αποθετήρια, χρησιμοποιώντας λέξεις-κλειδιά στην περιγραφή τους.

Όταν δεν γνωρίζουμε το όνομα του πακέτου που αναζητούμε, μπορούμε να χρησιμοποιήσουμε την εντολή dnf search με λέξεις-κλειδιά, όπως στο παρακάτω παράδειγμα:

[user@host ~]$ dnf search "text editor"

Last metadata expiration check: 4:52:16 ago on Wed 11 Dec 2024 07:39:17 AM EET.

======================== Summary Matched: text editor ========================

emacs.x86\_64 : GNU Emacs text editor

emacs-lucid.x86\_64 : GNU Emacs text editor with LUCID toolkit X support

emacs-nox.x86\_64 : GNU Emacs text editor without X support

gedit.x86\_64 : Text editor for the GNOME desktop

nano.x86\_64 : A small text editor

sed.x86\_64 : A GNU stream text editor

Από τα αποτελέσματα, οι διαθέσιμοι επεξεργαστές κειμένου περιλαμβάνουν τους Emacs, Gedit και Nano. Το sed θεωρείται περισσότερο εργαλείο επεξεργασίας κειμένου παρά πλήρης επεξεργαστής.

**Σενάριο 2: Υπολογισμός πλήθους εγκατεστημένων πακέτων λογισμικού**

Στόχος: Να υπολογίσουμε, κατά προσέγγιση, το πλήθος των πακέτων λογισμικού που είναι εγκατεστημένα στο σύστημά μας.

Χρησιμοποιούμε την εντολή dnf list installed για να εμφανίσουμε τα εγκατεστημένα πακέτα. Στη συνέχεια, μπορούμε να μετρήσουμε τον αριθμό των γραμμών της εξόδου με την εντολή wc -l:

[user@host ~]$ dnf list installed | wc -l

1318

Το σύστημά μας έχει περίπου 1.300 εγκατεστημένα πακέτα λογισμικού.

**Σενάριο 3: Ανίχνευση πακέτων εκτός αποθετηρίων**

Στόχος: Να εντοπίσουμε πακέτα λογισμικού που έχουν εγκατασταθεί τοπικά ή με διαφορετικό τρόπο (π.χ., μεταγλώττιση από πηγαίο κώδικα) και δεν ανήκουν σε κάποιο αποθετήριο.

Η εντολή dnf list extras εμφανίζει όλα τα πακέτα που είναι εγκατεστημένα στο σύστημα αλλά δεν βρίσκονται σε κανένα αποθετήριο:

[user@host ~]$ dnf list extras

Last metadata expiration check: 5:18:24 ago on Wed 11 Dec 2024 07:39:17 AM EET.

Στο συγκεκριμένο σύστημα, δεν υπάρχουν εγκατεστημένα πακέτα εκτός αποθετηρίων.

**Σενάριο 4: Έλεγχος ενημερώσεων για συγκεκριμένο πακέτο λογισμικού**

Στόχος: Να ελέγξουμε αν υπάρχει διαθέσιμη ενημέρωση για ένα πακέτο λογισμικού (π.χ., firefox).

Χρησιμοποιούμε την εντολή dnf list updates για να ελέγξουμε αν υπάρχουν ενημερώσεις για το συγκεκριμένο πακέτο:

[user@host ~]$ dnf list updates firefox

Last metadata expiration check: 5:41:06 ago on Wed 11 Dec 2024 07:39:17 AM EET.

Error: No matching Packages to list

Η απουσία αποτελεσμάτων δείχνει ότι το πακέτο firefox βρίσκεται ήδη στην τελευταία διαθέσιμη έκδοση.

### Εργασία 3: Πληροφορίες για πακέτο λογισμικού

Σε αυτή την εργασία, θα εξερευνήσουμε διαφορετικά σενάρια χρήσης για την αναζήτηση πληροφοριών σχετικά με πακέτα λογισμικού στο Linux.

**Σενάριο 1: Εύρεση πληροφοριών για συγκεκριμένο πακέτο**

Στόχος: Να ανακτήσουμε πληροφορίες για ένα πακέτο λογισμικού για το οποίο γνωρίζουμε το όνομα του (yara), χωρίς να γνωρίζουμε αν αυτό είναι εγκατεστημένο είτε όχι.

Για την αναζήτηση πληροφοριών σχετικά με ένα συγκεκριμένο πακέτο, χρησιμοποιούμε την εντολή dnf info.

[user@host ~]$ dnf info yara

Η εντολή εμφανίζει λεπτομέρειες όπως περιγραφή, έκδοση, μέγεθος και το αποθετήριο στο οποίο ανήκει το πακέτο.

**Σενάριο 2: Εμφάνιση των περιεχομένων ενός πακέτου**

Στόχος: Να δούμε τα αρχεία που περιλαμβάνονται σε ένα πακέτο λογισμικού (green-boot), είτε αυτό είναι εγκατεστημένο στο σύστημα είτε βρίσκεται σε αποθετήρια.

Χρησιμοποιούμε την εντολή dnf repoquery -l για την εμφάνιση των περιεχομένων.

[user@host ~]$ dnf repoquery -l greenboot

Last metadata expiration check: 0:03:44 ago on Wed 11 Dec 2024 06:49:00 PM EET.

/etc/greenboot

/etc/greenboot/check

/etc/greenboot/check/required.d

/etc/greenboot/check/wanted.d

/etc/greenboot/green.d

[τα αποτελέσματα έχουν περικοπεί για λόγους οικονομίας χώρου]

**Σενάριο 3: Εύρεση του πακέτου που παρέχει μια εντολή**

Στόχος: Να βρούμε ποιο πακέτο λογισμικού περιλαμβάνει μια συγκεκριμένη εντολή (pip3) ή αρχείο στο σύστημα.

Χρησιμοποιούμε την εντολή dnf provides για να προσδιορίσουμε το πακέτο που περιέχει την εντολή ή το αρχείο.

[user@host ~]$ dnf provides pip3

Last metadata expiration check: 0:11:40 ago on Wed 11 Dec 2024 06:49:00 PM EET.

python3-pip-21.3.1-1.el9.noarch : A tool for installing and managing Python3

: packages

Repo : appstream

Matched from:

Filename : /usr/bin/pip3

Το πακέτο python3-pip περιέχει την εντολή pip3 και μπορεί να εγκατασταθεί με την εντολή:

[user@host ~]$ sudo dnf install python3-pip

**Σενάριο 4: Έλεγχος ακεραιότητας ενός πακέτου**

Στόχος: Να ελέγξουμε την ακεραιότητα ενός αρχείου πακέτου .rpm για να επιβεβαιώσουμε ότι δεν έχει αλλοιωθεί.

Αρχικά, εγκαθιστούμε το αποθετήριο EPEL για να αποκτήσουμε πρόσβαση σε περισσότερα πακέτα:

[user@host ~]$ sudo dnf install epel-release

Επιβεβαιώνουμε ότι το αποθετήριο EPEL είναι ενεργό:

[user@host ~]$ dnf repolist | grep epel

epel Extra Packages for Enterprise Linux 9 - x86\_64

epel-cisco-openh264 Extra Packages for Enterprise Linux 9 openh264 (From Cisco) - x86\_64

Κατεβάζουμε το πακέτο htop από το αποθετήριο EPEL χωρίς να το εγκαταστήσουμε:

[user@host ~]$ dnf download htop

Last metadata expiration check: 0:13:19 ago on Thu 12 Dec 2024 08:33:20 AM EET.

htop-3.3.0-1.el9.x86\_64.rpm 478 kB/s | 198 kB 00:00

Χρησιμοποιούμε την εντολή rpm -K για να ελέγξουμε την ακεραιότητα του αρχείου:

[user@host ~]$ rpm -K htop-3.3.0-1.el9.x86\_64.rpm

htop-3.3.0-1.el9.x86\_64.rpm: digests SIGNATURES OK

Για να δοκιμάσουμε την ανίχνευση αλλοίωσης, μπορούμε να τροποποιήσουμε εσκεμμένα το αρχείο και να εκτελέσουμε ξανά τον έλεγχο:

[user@host ~]$ echo "hello" >> htop-3.3.0-1.el9.x86\_64.rpm

[user@host ~]$ rpm -K htop-3.3.0-1.el9.x86\_64.rpm

htop-3.3.0-1.el9.x86\_64.rpm: DIGESTS SIGNATURES NOT OK

Ο έλεγχος δείχνει ότι το αρχείο έχει αλλοιωθεί.

### Εργασία 4: Διαχείριση ομάδων πακέτων λογισμικού

Σε αυτό το εργαστήριο, θα εξετάσουμε βασικές λειτουργίες διαχείρισης ομάδων πακέτων λογισμικού στο Linux. Οι ομάδες πακέτων είναι συλλογές σχετικών πακέτων που εξυπηρετούν έναν κοινό σκοπό, όπως γραφικά περιβάλλοντα, εργαλεία ασφαλείας ή διαχείρισης συστήματος.

**Σενάριο 1: Εμφάνιση όλων των διαθέσιμων ομάδων πακέτων λογισμικού**

Στόχος: Να εμφανίσουμε όλες τις διαθέσιμες ομάδες πακέτων λογισμικού, είτε αυτές είναι ήδη εγκατεστημένες είτε διαθέσιμες για εγκατάσταση.

Χρησιμοποιούμε την εντολή dnf group list για την εμφάνιση των ομάδων πακέτων.

[user@host ~]$ dnf group list hidden

Last metadata expiration check: 0:28:38 ago on Wed 11 Dec 2024 06:49:00 PM EET.

Available Environment Groups:

Server with GUI

Server

Minimal Install

[η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου]

**Σενάριο 2: Εγκατάσταση ομάδας πακέτων λογισμικού**

Στόχος: Να εγκαταστήσουμε μια ομάδα πακέτων λογισμικού (Security Tools) χρησιμοποιώντας είτε τη λειτουργία ομάδων του dnf είτε την τυπική εντολή εγκατάστασης με χρήση ειδικής σύνταξης.

Η εγκατάσταση μπορεί να γίνει με την εντολή dnf group install ή με την εντολή dnf install και προσθέτοντας τον χαρακτήρα @ πριν από το όνομα της ομάδας.

[user@host ~]$ sudo dnf install @"Security Tools"

Last metadata expiration check: 0:00:04 ago on Wed 11 Dec 2024 07:21:42 PM EET.

Dependencies resolved.

==============================================================================

Package Arch Version Repository Size

==============================================================================

Installing group/module packages:

scap-security-guide noarch 0.1.75-1.el9.rocky.1.1 appstream 1.7 M

[η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου]

Η εντολή θα εγκαταστήσει όλα τα απαραίτητα πακέτα της ομάδας, μαζί με τις εξαρτήσεις τους.

**Σενάριο 3: Εμφάνιση πληροφοριών για τα περιεχόμενα μιας ομάδας**

Στόχος: Να εμφανίσουμε τα πακέτα λογισμικού που περιλαμβάνονται σε μια συγκεκριμένη ομάδα (Headless Management).

Χρησιμοποιούμε την εντολή dnf group info για να δούμε λεπτομέρειες σχετικά με τα πακέτα μιας ομάδας.

[user@host ~]$ dnf group info "Headless Management"

Last metadata expiration check: 0:36:21 ago on Wed 11 Dec 2024 06:49:00 PM EET.

Group: Headless Management

Description: Tools for managing the system without an attached graphical console.

Mandatory Packages:

PackageKit

cockpit-bridge

[η έξοδος έχει περικοπεί για λόγους οικονομίας χώρου]

Η έξοδος περιλαμβάνει τη λίστα των πακέτων που ανήκουν στη συγκεκριμένη ομάδα.

**Σενάριο 4: Εμφάνιση εγκατεστημένων ομάδων πακέτων λογισμικού**

Στόχος: Να δούμε ποιες ομάδες πακέτων λογισμικού είναι ήδη εγκατεστημένες στο σύστημά μας.

Χρησιμοποιούμε την εντολή dnf group list installed για την εμφάνιση των εγκατεστημένων ομάδων πακέτων λογισμικού.

[user@host ~]$ dnf group list installed

### Εργασία 5: Ρυθμίσεις του DNF και των αποθετηρίων

Σε αυτό το εργαστήριο, θα εξετάσουμε τη ρύθμιση του DNF, με στόχο την προσαρμογή της λειτουργίας του στις ανάγκες του συστήματος.

Στόχος: Να ενεργοποιήσουμε και να απενεργοποιήσουμε ένα αποθετήριο (plus) το οποίο γνωρίζει ήδη ο DNF.

Για ενεργοποίηση του αποθετηρίου plus χρησιμοποιούμε την εντολή:

[user@host ~]$ sudo dnf config-manager --set-enabled plus

Επαλήθευση ενεργοποίησης:

[user@host ~]$ dnf repolist

repo id repo name

appstream Rocky Linux 9 - AppStream

baseos Rocky Linux 9 - BaseOS

extras Rocky Linux 9 - Extras

plus Rocky Linux 9 - Plus

Απενεργοποίηση του αποθετηρίου plus:

[user@host ~]$ sudo dnf config-manager --set-disabled plus

Επαλήθευση απενεργοποίησης:

[user@host ~]$ dnf repolist

repo id repo name

appstream Rocky Linux 9 - AppStream

baseos Rocky Linux 9 - BaseOS

extras Rocky Linux 9 - Extras

Σενάριο 2: Καθαρισμός των προσωρινών αρχείων (cache) του DNF

Στόχος: Να διαγράψουμε τα προσωρινά αρχεία (cache) που διατηρεί το DNF και σχετίζονται με τις πληροφορίες αποθετηρίων.

Η εντολή για την εκκαθάριση της προσωρινής μνήμης είναι η dnf clean. Θα χρησιμοποιήσουμε την επιλογή dbcache γιατί θέλουμε τη διαγραφή της βάσης δεδομένων cache.

[user@host ~]$ sudo dnf clean dbcache

8 files removed

[user@host ~]$

Μετά τον καθαρισμό της προσωρινής μνήμης, η πρώτη αναζήτηση στα αποθετήρια ενδέχεται να διαρκέσει περισσότερο, καθώς το DNF θα χρειαστεί να κατεβάσει ξανά τις πληροφορίες.

**Σενάριο 3: Δημιουργία και προσαρμογή αποθετηρίου μέσω αρχείου ρυθμίσεων**

Στόχος: Να δημιουργήσουμε ή να τροποποιήσουμε ένα αρχείο ρύθμισης αποθετηρίου στον κατάλογο /etc/yum.repos.d/.

Τα αποθετήρια στο DNF καθορίζονται μέσω αρχείων .repo στον κατάλογο /etc/yum.repos.d/. Κάθε αρχείο .repo περιέχει τμήματα (sections) με πληροφορίες για το αποθετήριο, όπως το URL του, το αν είναι ενεργοποιημένο, και τις επιλογές GPG.

**Δημιουργία αρχείου ρύθμισης:**

Χρησιμοποιούμε έναν επεξεργαστή κειμένου για να δημιουργήσουμε ένα νέο αρχείο αποθετηρίου, π.χ., custom.repo.

[user@host ~]$ sudo vi /etc/yum.repos.d/custom.repo

**Προσθήκη περιεχομένου αποθετηρίου:**

Το παρακάτω είναι ένα δείγμα περιεχομένου που μπορεί να προστεθεί:

[custom-repo]

name=Custom Repository

baseurl=http://example.com/repo/

enabled=1

gpgcheck=1

gpgkey=http://example.com/repo/RPM-GPG-KEY

Αποθηκεύουμε το αρχείο και κλείνουμε τον επεξεργαστή.

Ελέγχουμε αν το αποθετήριο είναι διαθέσιμο μέσω της εντολής:

[user@host ~]$ dnf repolist

## Ρύθμιση της πρόσβασης στο σύστημα των αρχείων

Μία από τις βασικές σχεδιαστικές αρχές του λειτουργικού συστήματος Linux είναι η αναπαράσταση όλων των στοιχείων του υπολογιστικού συστήματος ως αρχεία. Το ίδιο ισχύει και για τις συσκευές αποθήκευσης, οι οποίες ονομάζονται και σαν συσκευές τύπου **block**.

### Συσκευές τύπου Block

Οι συσκευές αποθήκευσης χαρακτηρίζονται ως συσκευές τύπου **block** επειδή επιτρέπουν την πρόσβαση σε δεδομένα με τη μορφή blocks, δηλαδή σε σταθερού μεγέθους τμήματα πληροφορίας. Στο Linux, κάθε συσκευή block αναπαρίσταται από ένα αρχείο στο σύστημα αρχείων. Μπορούμε να δούμε ένα τέτοιο αρχείο με την παρακάτω εντολή:

[root@host dev]# ls -l /dev/sda

brw-rw----. 1 root disk 8, 0 Nov 21 07:27 /dev/sda

Στην έξοδο της εντολής ls -l, ο πρώτος χαρακτήρας είναι το γράμμα b, που υποδηλώνει ότι πρόκειται για συσκευή τύπου block.

**Ο Ρόλος της Υπηρεσίας udev**

Στο Linux, η διαχείριση των συσκευών τύπου block γίνεται μέσω της υπηρεσίας **udev**, η οποία λειτουργεί στο παρασκήνιο. Η υπηρεσία αυτή παρακολουθεί τις συνδεδεμένες συσκευές και δημιουργεί τα αντίστοιχα αρχεία στο φάκελο /dev όταν ανιχνεύει νέες συσκευές. Μπορούμε να επαληθεύσουμε τη λειτουργία της υπηρεσίας **udev** με την εντολή:

[root@host dev]# ps -e | grep udev

736 ? 00:00:00 systemd-udevd

Η υπηρεσία **udev** εφαρμόζει συγκεκριμένους κανόνες για την ονομασία των αρχείων που αντιστοιχούν σε συσκευές τύπου block. Το πρόθεμα του αρχείου εξαρτάται από το είδος της συσκευής, όπως φαίνεται στον παρακάτω πίνακα:

|  |  |
| --- | --- |
| **Τύπος Συσκευής** | **Αρχικό Πρόθεμα Ονόματος Αρχείου** |
| Σκληρός Δίσκος IDE | hd |
| Σκληρός Δίσκος SCSI/SATA | sd |
| Σκληρός Δίσκος NVMe | nvme |
| CD-ROM | sr |
| Δισκέτα (Floppy Disk) | fd |

**Ονοματοδοσία Συσκευών Block**

Μετά το αρχικό πρόθεμα, το όνομα του αρχείου περιλαμβάνει έναν χαρακτήρα που αναπαριστά τον αύξοντα αριθμό της συσκευής (π.χ., a για την πρώτη συσκευή, b για τη δεύτερη, κ.ο.κ.). Για παράδειγμα, το αρχείο /dev/sda αντιστοιχεί στον πρώτο σκληρό δίσκο τύπου SCSI/SATA.

**Ιδιαιτερότητες Συσκευών NVMe**

Οι δίσκοι NVMe διαθέτουν διαφορετική ονοματοδοσία λόγω της υποστήριξης namespaces, μιας τεχνολογίας που επιτρέπει τη δημιουργία πολλών «λογικών» δίσκων από έναν φυσικό δίσκο NVMe. Το όνομα του πρώτου λογικού δίσκου είναι συνήθως nvme0n1. Ένα παράδειγμα αρχείου που αναπαριστά έναν NVMe δίσκο παρουσιάζεται παρακάτω:

[root@host dev]# ls -l nvme0n1

brw-rw----. 1 root disk 259, 0 Nov 21 06:45 nvme0n1

Κατατμήσεις (partitions)

Οι **κατατμήσεις** (**partitions**) επιτρέπουν τη διαίρεση μιας φυσικής συσκευής αποθήκευσης (π.χ., σκληρού δίσκου) σε περισσότερες «εικονικές» μονάδες αποθήκευσης. Το λειτουργικό σύστημα βλέπει την κάθε κατάτμηση ως ξεχωριστή συσκευή, επιτρέποντας έτσι την καλύτερη αξιοποίηση του διαθέσιμου αποθηκευτικού χώρου και την αύξηση της ασφάλειας και της ευελιξίας.

**Τύποι Partition Tables: MBR και GPT**

Υπάρχουν δύο βασικές προσεγγίσεις για τη δημιουργία και την οργάνωση των κατατμήσεων σε ένα δίσκο:

1. **Master Boot Record (MBR):**
   * Υποστηρίζει δίσκους έως 2 Terabytes.
   * Επιτρέπει τη δημιουργία έως 4 **primary partitions**.
   * Μία από αυτές μπορεί να είναι **extended**, η οποία με τη σειρά της μπορεί να περιέχει πολλά **logical partitions**.
2. **GUID Partition Table (GPT):**
   * Υποστηρίζει δίσκους μεγαλύτερους από 2TB.
   * Επιτρέπει τη δημιουργία έως και 128 κατατμήσεων χωρίς την ανάγκη extended partitions.
   * Χρησιμοποιείται ευρέως σε σύγχρονα συστήματα λόγω της αυξημένης ευελιξίας και αξιοπιστίας.

Στο Linux, η διαχείριση των κατατμήσεων γίνεται με εργαλεία όπως τα fdisk, gdisk και parted. Όλα αυτά απαιτούν δικαιώματα διαχειριστή για να λειτουργήσουν. Στην παρακάτω ενότητα, θα επικεντρωθούμε στη δημιουργία **MBR** κατατμήσεων με τη χρήση της εντολής fdisk.

**Δημιουργία MBR Κατατμήσεων με το fdisk**

Το fdisk είναι ένα από τα παλαιότερα εργαλεία διαχείρισης κατατμήσεων στο Linux, διαθέσιμο σε όλες τις διανομές. Η διαδικασία δημιουργίας κατατμήσεων περιλαμβάνει τα εξής βήματα:

1. Είσοδος στο περιβάλλον του fdisk: Εκτελούμε την εντολή fdisk ακολουθούμενη από το όνομα της συσκευής που θέλουμε να διαχειριστούμε:

[user@host ~]$ sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.37.4).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table.

Created a new DOS disklabel with disk identifier 0xb85d3b98.

Η έξοδος της εντολή fdisk μας ενημερώνει ότι στον συγκεκριμένο δίσκο δεν υπάρχει αυτή την στιγμή πίνακας κατατμήσεων.

Το περιβάλλον του fdisk υποστηρίζει διάφορες εντολές. Στον παρακάτω πίνακα παρουσιάζονται οι πιο χρήσιμες:

|  |  |  |
| --- | --- | --- |
| **Εντολή** | **Συντόμευση** | **Λειτουργία** |
| New | n | Δημιουργία νέας κατάτμησης |
| Delete | d | Διαγραφή υπάρχουσας κατάτμησης |
| Print | p | Εμφάνιση του πίνακα κατατμήσεων |
| Free | f | Εμφάνιση μη κατανεμημένου χώρου |
| Write | w | Αποθήκευση αλλαγών και έξοδος |
| Quit | q | Έξοδος χωρίς αποθήκευση αλλαγών |

2. Δημιουργία Partition: Για να δημιουργήσουμε μια νέα κατάτμηση, πληκτρολογούμε n. Στη συνέχεια, ακολουθούμε τα βήματα που ζητάει το fdisk:

Επιλογή Τύπου Partition: Το fdisk ρωτά αν θέλουμε να δημιουργήσουμε ένα primary ή extended partition. Επιλέγουμε p για primary:

Command (m for help): n

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p):

Αριθμός Partition: Επιλέγουμε έναν αριθμό από 1 έως 4 (για primary partitions). Ξεκινάμε με το 1.

Partition number (1-4, default 1): 1

Πρώτος Τομέας (Sector): Ορίζουμε τον πρώτο τομέα για το partition. Από προεπιλογή, το fdisk προτείνει το πρώτο διαθέσιμο (συνήθως 2048):

First sector (2048-20971519, default 2048): 2048

Τελευταίος Τομέας (Sector) ή Μέγεθος: Μπορούμε να δώσουμε έναν τομέα ή ένα συγκεκριμένο μέγεθος για το partition (π.χ., +6G για 6GB):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519, default 20971519): +6G

Created a new partition 1 of type 'Linux' and of size 6 GiB.

Command (m for help):

Αφού ολοκληρώσουμε αυτά τα βήματα, το partition δημιουργείται και εμφανίζεται με την εντολή p:

Command (m for help): p

Disk /dev/sda: 10 GiB, 10737418240 bytes, 20971520 sectors

Disk model: VMware Virtual S

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xb85d3b98

Device Boot Start End Sectors Size Id Type

/dev/sda1 2048 12584959 12582912 6G 83 Linux

Command (m for help):

Η διαδικασία δημιουργίας κατατμήσεων στο fdisk δεν αποθηκεύει αυτόματα τις αλλαγές στον πίνακα κατατμήσεων. Όλες οι ενέργειες που πραγματοποιείτε κατά τη διάρκεια της συνεδρίας του fdisk εκτελούνται μόνο στη μνήμη. Για να αποθηκευτούν οι αλλαγές, πρέπει να χρησιμοποιήσουμε την εντολή w (write).

**Δημιουργία GPT κατατμήσεων με την parted**

Το εργαλείο parted είναι μία ευέλικτη και ισχυρή εντολή που επιτρέπει τη δημιουργία και τη διαχείριση κατατμήσεων σε δίσκους που σχεδιάστηκε από την αρχή για να υποστηρίζει τόσο το MBR όσο και το GPT.

Για να ξεκινήσουμε, εκτελούμε την εντολή parted με δικαιώματα διαχειριστή, καθορίζοντας τη συσκευή αποθήκευσης που θέλουμε να διαχειριστούμε:

[user@host ~]$ sudo parted /dev/sda

Μετά την εκτέλεση της εντολής, εισερχόμαστε στο διαδραστικό περιβάλλον της parted:

GNU Parted 3.5

Using /dev/sda

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted)

Όπως και στο fdisk, μπορούμε να εκτελέσουμε συγκεκριμένες εντολές για τη διαχείριση των κατατμήσεων. Οι βασικές εντολές παρουσιάζονται παρακάτω:

|  |  |
| --- | --- |
| **Εντολή** | **Λειτουργικότητα** |
| help | Εμφάνιση όλων των διαθέσιμων εντολών |
| mklabel | Δημιουργία partition table |
| mkpart | Δημιουργία νέας κατάτμησης |
| print | Εμφάνιση του partition table |
| select | Επιλογή δίσκου ή partition |
| quit | Έξοδος από την parted |

Δημιουργία GPT Partition Table: Για να δημιουργήσουμε ένα νέο GPT partition table, χρησιμοποιούμε την εντολή mklabel (ή mktable):

(parted) mklabel

New disk label type? gpt

(parted)

Η parted μας ζητάει να καθορίσουμε τον τύπο του partition table. Εδώ, επιλέγουμε gpt. Με αυτήν την ενέργεια, διαγράφονται όλες οι υπάρχουσες κατατμήσεις στον δίσκο.

Δημιουργία Νέου Partition: Για να δημιουργήσουμε ένα νέο partition, χρησιμοποιούμε την εντολή mkpart. Κατά την εκτέλεση, η parted ζητάει τις ακόλουθες πληροφορίες:

1. **Όνομα του partition** (προαιρετικό).
2. **Τύπο συστήματος αρχείων** (filesystem). Για παράδειγμα, ext4. Περισσότερες λεπτομέρειες για τα συστήματα αρχείων θα συζητηθούν στην συνέχεια.
3. **Σημείο εκκίνησης του partition** (π.χ., ποσοστό ή συγκεκριμένο μέγεθος).
4. **Σημείο τερματισμού του partition** (π.χ., ποσοστό ή συγκεκριμένο μέγεθος).

(parted) mkpart

Partition name? []?

File system type? [ext2]? ext4

Start? 0%

End? 60%

(parted)

Στο παραπάνω παράδειγμα το partition ξεκινά από το 0% και τελειώνει στο 60% του συνολικού χώρου του δίσκου.

Με την εντολή print (ή p), μπορούμε να δούμε την τρέχουσα κατάσταση του partition table:

(parted) p

Model: ATA VMware Virtual S (scsi)

Disk /dev/sda: 10.7GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Disk Flags:

Number Start End Size File system Name Flags

1 1049kB 6442MB 6441MB ext4

(parted)

Σε αντίθεση με το fdisk, η parted αποθηκεύει αυτόματα τις αλλαγές στο partition table. Δεν χρειάζεται κάποια επιπλέον εντολή για εγγραφή. Αυτό μειώνει την πιθανότητα απώλειας δεδομένων σε περίπτωση που ξεχάσουμε να αποθηκεύσουμε τις αλλαγές μας. Τέλος υπάρχει και το γραφικό εργαλείο gparted, το οποίο βασίζεται στο parted. Παρέχει ένα εύχρηστο interface για τη διαχείριση κατατμήσεων, με οπτική απεικόνιση του δίσκου και εύκολη εκτέλεση των ίδιων λειτουργιών.

### Συστήματα αρχείων (filesystems)

Όταν αποθηκεύουμε ένα αρχείο σε μια συσκευή αποθήκευσης, δεν καταγράφονται μόνο τα δεδομένα του αρχείου, αλλά και πληροφορίες που το συνοδεύουν. Οι πληροφορίες αυτές περιλαμβάνουν, μεταξύ άλλων, τον ιδιοκτήτη του αρχείου, τη στιγμή δημιουργίας και τελευταίας τροποποίησης, τα δικαιώματα πρόσβασης, και την τοποθεσία των δεδομένων του αρχείου (σε συγκεκριμένα sectors του δίσκου). Αυτές οι πληροφορίες ονομάζονται **μεταδεδομένα**.

Η οργάνωση των μεταδεδομένων, καθώς και των δεδομένων του ίδιου του αρχείου, καθορίζεται από το **σύστημα αρχείων** (**filesystem**) του δίσκου. Το σύστημα αρχείων παρέχει μια λογική δομή που επιτρέπει στο λειτουργικό σύστημα να αποθηκεύει, να ανακτά και να διαχειρίζεται δεδομένα αποτελεσματικά. Μπορεί να εφαρμοστεί είτε σε έναν ολόκληρο δίσκο είτε σε ένα συγκεκριμένο partition. Μάλιστα, κάθε partition ενός δίσκου μπορεί να έχει διαφορετικό σύστημα αρχείων, ανάλογα με τις ανάγκες του χρήστη.

**Υποστηριζόμενα Συστήματα Αρχείων στο Linux**

Το Linux υποστηρίζει μια ευρεία γκάμα συστημάτων αρχείων, το καθένα από τα οποία έχει σχεδιαστεί για διαφορετικές ανάγκες:

* **Ext4** (**Fourth Extended Filesystem**): Ένα από τα πιο διαδεδομένα συστήματα αρχείων στο Linux, κατάλληλο για γενική χρήση.
* **XFS**: Ένα υψηλής απόδοσης σύστημα αρχείων, ιδανικό για μεγάλα αρχεία και server workloads.
* **Btrfs** (**B-Tree Filesystem**): Προσφέρει προηγμένα χαρακτηριστικά, όπως snapshots και υποστήριξη RAID.
* **F2FS** (**Flash-Friendly File System**): Βελτιστοποιημένο για συσκευές αποθήκευσης τύπου flash, όπως SSDs και κάρτες μνήμης.

**Δημιουργία Συστήματος Αρχείων με την mkfs**

Η δημιουργία ενός συστήματος αρχείων γίνεται με τις εντολές της οικογένειας mkfs. Για να δείτε όλες τις διαθέσιμες επιλογές, πληκτρολογήστε την εντολή mkfs και πατήστε δύο φορές το πλήκτρο Tab. Ο φλοιός (shell) θα εμφανίσει όλες τις εντολές που ξεκινούν με mkfs, όπως φαίνεται στο παρακάτω παράδειγμα:

[user@host ~]$ mkfs

mkfs mkfs.ext2 mkfs.ext4 mkfs.minix mkfs.vfat

mkfs.cramfs mkfs.ext3 mkfs.fat mkfs.msdos mkfs.xfs

[user@host ~]$ mkfs

Για παράδειγμα, για να εφαρμόσουμε το σύστημα αρχείων ext4 στο partition /dev/sda1, χρησιμοποιούμε την ακόλουθη εντολή:

[user@host ~]$ sudo mkfs.ext4 /dev/sda1

[sudo] password for user:

mke2fs 1.46.5 (30-Dec-2021)

Creating filesystem with 1572608 4k blocks and 393216 inodes

Filesystem UUID: 7b460520-287a-45f4-9043-b73bdf149106

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done

Writing inode tables: done

Creating journal (16384 blocks): done

Writing superblocks and filesystem accounting information: done

[user@host ~]$

Με τη δημιουργία ενός συστήματος αρχείων σε έναν δίσκο ή partition, δημιουργείται ένας **τόμος** (**volume**), τον οποίο το λειτουργικό σύστημα μπορεί να αναγνωρίσει και να χρησιμοποιήσει για την αποθήκευση δεδομένων.

**Προβολή Συστήματος Αρχείων με την df**

Για να εμφανίσουμε τους τόμους που είναι ορισμένοι στο σύστημα, χρησιμοποιούμε την εντολή df:

[user@host ~]$ df

Filesystem 1K-blocks Used Available Use% Mounted on

devtmpfs 4096 0 4096 0% /dev

tmpfs 1857308 0 1857308 0% /dev/shm

tmpfs 742924 11360 731564 2% /run

/dev/mapper/rl\_localhost--live-root 17756160 6052024 11704136 35% /

tmpfs 1857308 200 1857108 1% /tmp

/dev/nvme0n1p1 983040 430596 552444 44% /boot

tmpfs 371460 120 371340 1% /run/user/1000

Η παραπάνω έξοδος δείχνει τα συστήματα αρχείων που χρησιμοποιούνται, μαζί με πληροφορίες όπως η διαθέσιμη χωρητικότητα και το **σημείο προσάρτησης** (**mount point**). Μπορούμε να αποκρύψουμε ορισμένα συστήματα αρχείων, όπως τα εικονικά **tmpfs**, χρησιμοποιώντας την επιλογή -x:

[user@host ~]$ df -x tmpfs --output=source,fstype,pcent,target

Filesystem Type Use% Mounted on

devtmpfs devtmpfs 0% /dev

/dev/mapper/rl\_localhost--live-root xfs 35% /

/dev/nvme0n1p1 xfs 44% /boot

[user@host ~]$

Στο παραπάνω παράδειγμα, φιλτράραμε τα συστήματα αρχείων tmpfs και επιλέξαμε συγκεκριμένα πεδία εξόδου με την επιλογή --output.

### Προσάρτηση συσκευών αποθήκευσης

Το τελευταίο βήμα για τη χρήση μιας αποθηκευτικής συσκευής στο Linux είναι η **προσάρτησή** της (**mount**) στο ιεραρχικό «δέντρο» καταλόγων του συστήματος. Στο Linux, όλα τα αρχεία που είναι προσβάσιμα είναι οργανωμένα σε ένα ενιαίο δέντρο καταλόγων, που ξεκινά από τον ριζικό κατάλογο (root catalog, /). Για να γίνει αυτό, πρέπει να αντιστοιχίσουμε έναν υποκατάλογο του πατρικού καταλόγου στη συσκευή αποθήκευσης που θέλουμε να προσαρτήσουμε. Αυτό το πετυχαίνουμε με την χρήση της εντολής mount με δικαιώματα διαχειριστή. Η γενική σύνταξη της εντολής mount είναι

mount [επιλογές] <συσκευή> <κατάλογος>

όπου <συσκευή> είναι η συσκευή αποθήκευσης που θέλουμε να προσαρτήσουμε (π.χ. /dev/sda1) ενώ <κατάλογος> ο κατάλογος του δέντρου στον οποίο θα προσαρτηθεί η συσκευή (π.χ. /mnt/labdata). Ο κατάλογος αυτός πρέπει να υπάρχει ήδη.

Για να προσαρτήσουμε τον τόμο /dev/sda1 στον κατάλογο /mnt/labdata, εκτελούμε τα εξής:

user@host ~]$ sudo mkdir /mnt/labdata

[user@host ~]$ sudo mount /dev/sda1 /mnt/labdata

Η προσάρτηση μπορεί να επαληθευτεί με την εντολή df:

Filesystem Type Use% Mounted on

devtmpfs devtmpfs 0% /dev

/dev/mapper/rl\_localhost--live-root xfs 35% /

/dev/nvme0n1p1 xfs 44% /boot

/dev/sda1 ext4 1% /mnt/labdata

Με την επιλογή -o μπορούμε να ορίσουμε πρόσθετες επιλογές για την εντολή mount, όπως την προσάρτηση μόνο για ανάγνωση (ro).

Για να διαγράψουμε την προσάρτηση μιας συσκευής, χρησιμοποιούμε την εντολή umount:

[user@host ~]$ sudo umount /mnt/labdata

**Το αρχείο /etc/fstab**

Οι προσαρτήσεις με την εντολή mount είναι προσωρινές και χάνονται μετά την επανεκκίνηση. Για να καταστούν μόνιμες, πρέπει να προσθέσουμε τις αντίστοιχες εγγραφές στο αρχείο **/etc/fstab**.

Το αρχείο /etc/fstab περιλαμβάνει μία γραμμή για κάθε τόμο που θέλουμε να προσαρτήσουμε. Ένα παράδειγμα μιας εγγραφής έχει την εξής μορφή:

/dev/sda1 /mnt/labdata ext4 defaults 0 0

Περιγραφή των πεδίων:

1. **Συσκευή**: Το όνομα της αποθηκευτικής συσκευής (π.χ. /dev/sda1).
2. **Κατάλογος**: Το σημείο προσάρτησης (π.χ. /mnt/labdata).
3. **Σύστημα Αρχείων**: Το σύστημα αρχείων του τόμου (π.χ. ext4).
4. **Επιλογές**: Ρυθμίσεις για την προσάρτηση (π.χ. defaults).
5. **Dump**: Ρύθμιση για τη χρήση του dump για δημιουργία αντιγράφων ασφαλείας (συνήθως 0).
6. **Pass**: Ρύθμιση για τον έλεγχο ακεραιότητας κατά την εκκίνηση. Η ρίζα (/) έχει τιμή 1, ενώ άλλοι τόμοι τιμή 2 ή 0 (για απενεργοποίηση).

Αν προσθέσουμε την παραπάνω εγγραφή στο αρχείο /etc/fstab στην συνέχεια πρέπει να ενημερώσουμε το σύστημα αλλά και την systemd με τις παρακάτω εντολές:

[user@host ~]$ sudo systemctl daemon-reload

[user@host ~]$ sudo mount -a

Η προσάρτηση μπορεί να επιβεβαιωθεί ξανά με την χρήση της εντολής df:

[user@host ~]$ df -x tmpfs

Filesystem 1K-blocks Used Available Use% Mounted on

devtmpfs 4096 0 4096 0% /dev

/dev/mapper/rl\_localhost--live-root 17756160 6039836 11716324 35% /

/dev/nvme0n1p1 983040 430596 552444 44% /boot

/dev/sr0 2076704 2076704 0 100% /run/media/user/Rocky-9-4-Workstation

/dev/sda1 6101600 24 5770672 1% /mnt/labdata

[user@host ~]$

### Το σύστημα Λογικών Τόμων (LVM)

Η παραδοσιακή διαχείριση κατατμήσεων παρουσιάζει σημαντικά μειονεκτήματα, όπως το στατικό μέγεθός τους και την ανάγκη για περίπλοκες διαδικασίες σε περίπτωση τροποποίησης του αποθηκευτικού χώρου. Το **Logical Volume Manager** (**LVM**) παρέχει μια δυναμική, ευέλικτη προσέγγιση που μειώνει το διαχειριστικό κόστος και βελτιώνει τη διαχείριση του αποθηκευτικού χώρου.

Το LVM λειτουργεί εισάγοντας νέες οντότητες που επιτρέπουν την οργάνωση και τη διαχείριση του αποθηκευτικού χώρου:

* **Physical Volumes** (**PV**)

Τα Physical Volumes αντιπροσωπεύουν τις φυσικές αποθηκευτικές μονάδες, όπως δίσκοι, κατατμήσεις ή συστήματα RAID. Για να συμμετάσχει μία κατάτμηση στο LVM, πρέπει να δηλωθεί ως Linux LVM Filesystem στον πίνακα κατατμήσεων (partition table).

* **Volume Groups** (**VG**)

Τα Volume Groups οργανώνουν τα Physical Volumes σε μία ενιαία μονάδα αποθηκευτικού χώρου. Αυτός ο χώρος μπορεί να κατανεμηθεί σε Logical Volumes. Κάθε Physical Volume χωρίζεται σε μικρά τμήματα, γνωστά ως Physical Extents (PE), τα οποία έχουν σταθερό μέγεθος (π.χ., 4 MiB) που καθορίζεται κατά τη δημιουργία του Volume Group.

* **Logical Volumes** (**LV**)

Τα Logical Volumes αποτελούν τη λογική μονάδα αποθηκευτικού χώρου που χρησιμοποιείται από το λειτουργικό σύστημα, λειτουργώντας όπως οι παραδοσιακές κατατμήσεις. Ωστόσο, μπορούν να διασπαστούν και να επεκταθούν δυναμικά, ακόμα και σε πολλαπλά Physical Volumes ή φυσικούς δίσκους.

**Πλεονεκτήματα του LVM**

Δυναμική διαχείριση χώρου: Η δυνατότητα επέκτασης και μείωσης του αποθηκευτικού χώρου χωρίς διακοπή της λειτουργίας.

Αυξημένη ευελιξία: Η δυνατότητα δημιουργίας Logical Volumes που εκτείνονται σε πολλαπλά Physical Volumes.

Απλοποίηση αντιγράφων ασφαλείας: Με τη χρήση εντολών όπως pvmove, είναι δυνατή η μετακίνηση δεδομένων σε νέους δίσκους, καθιστώντας ευκολότερη την αντικατάσταση αποθηκευτικών συσκευών.

**Λειτουργίες και Διαχείριση LVM**

Το LVM παρέχει ένα σύνολο εντολών για τη διαχείριση των Physical Volumes, Volume Groups και Logical Volumes. Οι πιο σημαντικές εντολές οργανώνονται στις εξής κατηγορίες:

|  |  |  |  |
| --- | --- | --- | --- |
| **Λειτουργία** | **Physical Volumes (PV)** | **Volume Groups (VG)** | **Logical Volumes (LV)** |
| Εμφάνιση | pvs | vgs | lvs |
| Εμφάνιση (αναλυτικά) | pvdisplay | vgdisplay | lvdisplay |
| Δημιουργία | pvcreate | vgcreate | lvcreate |
| Επέκταση |  | vgextend | lvextend |
| Μείωση |  | vgreduce | lvreduce |
| Διαγραφή | pvremove | vgremove | lvremove |
| Μετονομασία |  |  | lvrename |
| Μετακίνηση | pvmove |  |  |

### Αναζήτηση αρχείων στον δενδρικό κατάλογο με χρήση της locale

Στη διαχείριση συστημάτων Linux, συχνά προκύπτει η ανάγκη να εντοπιστεί ένα αρχείο ή κατάλογος με βάση χαρακτηριστικά όπως το όνομα, το μέγεθος, ή τα δικαιώματα. Η εντολή locate χρησιμοποιείται για τον γρήγορο εντοπισμό αρχείων ή καταλόγων με βάση το όνομά τους. Η ταχύτητά της οφείλεται στη χρήση μιας βάσης δεδομένων που περιλαμβάνει τις θέσεις των αρχείων στο σύστημα. Αυτή η βάση δεδομένων αποθηκεύεται συνήθως στο αρχείο /var/lib/mlocate/mlocate.db.

Η βασική σύνταξη της εντολής είναι:

**locate [επιλογές] <όνομα αρχείου>**

Όταν εκτελείται, η locate αναζητά το ζητούμενο όνομα αρχείου στη βάση δεδομένων αντί να σαρώνει το σύστημα αρχείων σε πραγματικό χρόνο. Ωστόσο, εάν η βάση δεδομένων δεν υπάρχει ή δεν είναι ενημερωμένη, η εντολή δεν μπορεί να επιστρέψει αποτελέσματα.

Σε περίπτωση που η εντολή locate εκτελεστεί για πρώτη φορά και το αρχείο βάσης δεδομένων δεν έχει δημιουργηθεί, το σύστημα επιστρέφει μήνυμα σφάλματος:

[user@localhost ~]$ locate myfile.log

locate: can not stat () `/var/lib/mlocate/mlocate.db': No such file or directory

Για να δημιουργήσουμε ή να ενημερώσουμε τη βάση δεδομένων, εκτελούμε την εντολή updatedb. Αυτή η διαδικασία πραγματοποιείται επίσης αυτόματα από το σύστημα σε τακτά χρονικά διαστήματα, ώστε να αντικατοπτρίζονται οι αλλαγές στο σύστημα αρχείων.

**Επιλογές της locate**

Η locate υποστηρίζει διάφορες επιλογές για πιο εξειδικευμένη αναζήτηση. Ο παρακάτω πίνακας παρουσιάζει μερικές από τις πιο χρήσιμες:

|  |  |
| --- | --- |
| **Επιλογή** | **Λειτουργικότητα** |
| -r | Χρήση regular expressions για πιο σύνθετα μοτίβα αναζήτησης. |
| -b | Αναζήτηση μόνο στο όνομα του αρχείου, αγνοώντας το όνομα των καταλόγων. |
| -e | Εμφάνιση μόνο αρχείων που υπάρχουν τη στιγμή της αναζήτησης. |
| -c | Εμφάνιση του αριθμού των αποτελεσμάτων. |
| -i | Αναζήτηση χωρίς διάκριση πεζών-κεφαλαίων. |
| -h | Εμφάνιση βοήθειας για τις επιλογές. |
| -n <αριθμός> | Εμφάνιση μόνο των πρώτων n αποτελεσμάτων. |

Για παράδειγμα αν θέλουμε να δούμε πόσα αρχεία περιέχουν τη λέξη "locate" στο όνομά τους, μπορούμε να χρησιμοποιήσουμε την επιλογή -bc:

[user@host ~]$ locate -bc locate

53

Το αποτέλεσμα υποδεικνύει ότι υπάρχουν 53 αρχεία που πληρούν το κριτήριο αναζήτησης.

### Αναζήτηση αρχείων στο δενδρικό κατάλογο με την χρήση της find

Η εντολή find είναι ένα από τα πιο ισχυρά εργαλεία που παρέχει το Linux για την αναζήτηση αρχείων και καταλόγων. Η λειτουργία της είναι ιδιαίτερα ευέλικτη, επιτρέποντας αναδρομική αναζήτηση σε συγκεκριμένες διαδρομές καταλόγων και εφαρμογή πολλών διαφορετικών κριτηρίων. Εκτός από την αναζήτηση, η find προσφέρει τη δυνατότητα εκτέλεσης εντολών στα αρχεία που ικανοποιούν τα κριτήρια αναζήτησης, καθιστώντας την απαραίτητη για την προηγμένη διαχείριση αρχείων.

**Σύνταξη της find**

Η βασική σύνταξη της find είναι η εξής:

find [επιλογές] {τοποθ\_αναζήτησης} {κριτ\_αναζήτησης} [ενέργειες]

**Τοποθ\_Αναζήτησης**: Μπορούμε να καθορίσουμε μία ή περισσότερες διαδρομές καταλόγων. Αν δεν ορίσουμε κάποια συγκεκριμένη διαδρομή, η αναζήτηση θα πραγματοποιηθεί στον τρέχοντα κατάλογο.

**Κριτ\_Αναζήτησης**: Καθορίζουμε τα χαρακτηριστικά που πρέπει να πληρούν τα αρχεία ή οι κατάλογοι.

**Ενέργειες**: Ορίζουμε εντολές που θα εφαρμόσουμε στα αρχεία που εντοπίζονται (π.χ., εμφάνιση, διαγραφή, εκτέλεση εντολής). Μερικές από τις πιο συνηθισμένες ενέργειες είναι:

|  |  |
| --- | --- |
| **Επιλογή** | **Ενέργεια** |
| -print | Εμφανίζει τη διαδρομή των αρχείων που εντοπίζονται. |
| -exec | Εκτελεί μια εντολή στα αποτελέσματα (απαιτεί τερματισμό με \;). |
| -ok | Όπως η -exec, αλλά ζητά επιβεβαίωση από τον χρήστη πριν από την εκτέλεση. |
| -delete | Διαγράφει τα αρχεία που εντοπίζονται. |
| -fprint | Αποθηκεύει τα αποτελέσματα σε ένα αρχείο που καθορίζουμε. |

Στην συνέχεια θα παρουσιάσουμε μερικά από τα κριτήρια αναζήτησης που μπορεί να υποστηρίξει η εντολή find.

Η παρουσίαση όλων των επιλογών της εντολής find θα απαιτούσε έναν ιδιαίτερα εκτενή όγκο πληροφοριών, ο οποίος υπερβαίνει τους σκοπούς του παρόντος υλικού. Για αναλυτικές πληροφορίες και παραδείγματα σχετικά με τη λειτουργία της, σας προτείνουμε να ανατρέξετε στη σελίδα του εγχειριδίου (manual page) της εντολής.

**Αναζήτηση με βάση το όνομα**

Μπορούμε να χρησιμοποιήσουμε την find για να αναζητήσουμε αρχεία ή φακέλους με βάση το όνομά τους. Η σύνταξη του κριτηρίου είναι:

-name <όνομα\_αναζήτησης>

Η find αναζητά αρχεία ή φακέλους που ταιριάζουν με το καθορισμένο όνομα στη διαδρομή αναζήτησης και στους υποκαταλόγους της. Μπορούμε να συμπεριλάβουμε μεταχαρακτήρες (wildcards) στο όνομα, τους οποίους είναι απαραίτητο να τοποθετήσουμε σε διπλά εισαγωγικά ώστε να αποφύγουμε την αλληλεπίδραση με τη λειτουργία του shell.

Για παράδειγμα αν θέλουμε να βρούμε όλα τα αρχεία στον κατάλογό μας (~) με το όνομα .bashrc, εκτελούμε:

[user@host ~]$ find ~ -name ".bashrc"

**Αντιμετώπιση μηνυμάτων σφάλματος**

Αν εκτελέσουμε την εντολή find σε καταλόγους στους οποίους δεν έχουμε δικαιώματα πρόσβασης, θα εμφανιστούν μηνύματα σφάλματος. Μπορούμε να αγνοήσουμε αυτά τα μηνύματα ανακατευθύνοντας την έξοδο λαθών στο /dev/null. Για παράδειγμα, για να αναζητήσουμε αρχεία PDF στον root κατάλογο:

[user@host ~]$ find / -name "\*.pdf" 2>/dev/null

**Αναζήτηση με αποκλεισμό κριτηρίων**

Αν θέλουμε να εξαιρέσουμε αρχεία που πληρούν ένα συγκεκριμένο κριτήριο, χρησιμοποιούμε τον χαρακτήρα !. Για παράδειγμα, μπορούμε να βρούμε όλα τα αρχεία στον κατάλογό μας (~) εκτός από αυτά με επέκταση .txt:

[user@host ~]$ find ~ ! -name "\*.txt"

Η find έχει μια μικρή ιδιαιτερότητα στη σύνταξη: οι επιλογές της, αν και αποτελούνται από λέξεις-κλειδιά, δεν απαιτούν διπλή παύλα (--) πριν από αυτές, αλλά μόνο μία παύλα (-).

**Αναζήτηση με βάση τον τύπο**

Η επιλογή -type χρησιμοποιείται για την αναζήτηση αρχείων συγκεκριμένου τύπου. Η περιγραφή τύπου είναι ένας χαρακτήρας που καθορίζει το είδος του αρχείου. Οι διαθέσιμες επιλογές είναι:

|  |  |
| --- | --- |
| **Χαρακτήρας** | **Τύπος Αρχείου** |
| F | Κανονικό αρχείο |
| D | Κατάλογος |
| L | Συμβολικός σύνδεσμος |
| C | Αρχείο συσκευής χαρακτήρα |
| B | Αρχείο συσκευής τύπου μπλοκ |
| P | Διοχέτευση (pipe) |
| S | Socket |

Παράδειγμα χρήσης:

Για να δούμε ποια από τα αρχεία στον κατάλογο /usr/bin είναι συμβολικοί σύνδεσμοι, μπορούμε να εκτελέσουμε:

[user@host ~]$ find /usr/bin -type l | head -3

/usr/bin/X

/usr/bin/apropos

/usr/bin/apropos.man-db

Σε αυτό το παράδειγμα, χρησιμοποιούμε την εντολή head για να περιορίσουμε την έξοδο στα πρώτα τρία αποτελέσματα.

**Αναζήτηση με βάση τον ιδιοκτήτη και την ομάδα ιδιοκτήτη**

Η επιλογή -user μας επιτρέπει να αναζητούμε αρχεία που ανήκουν σε έναν συγκεκριμένο χρήστη. Αντίστοιχα, η επιλογή -group χρησιμοποιείται για αναζήτηση με βάση την ομάδα ιδιοκτήτη. Επιπλέον, μπορούμε να χρησιμοποιήσουμε τα User IDs (UID) και Group IDs (GID) με τις επιλογές -uid και -gid, αντίστοιχα.

Παράδειγμα χρήσης:

Αν θέλουμε να αναζητήσουμε τα αρχεία που ανήκουν σε έναν συγκεκριμένο χρήστη user στον κατάλογο /var/spool, εκτελούμε:

[user@host ~]$ find /var/spool -user user 2>/dev/null

/var/spool/mail/user

Στο παραπάνω παράδειγμα:

* Αναζητούμε αρχεία που ανήκουν στον χρήστη user.
* Ανακατευθύνουμε τα μηνύματα σφάλματος στο /dev/null για να αποφύγουμε την εμφάνιση μηνυμάτων λάθους που προκύπτουν από καταλόγους στους οποίους δεν έχουμε δικαιώματα πρόσβασης.

**Αναζήτηση με βάση την ημερομηνία**

Η find παρέχει επιλογές που μας επιτρέπουν να φιλτράρουμε αρχεία σύμφωνα με την τελευταία πρόσβαση, τροποποίηση ή αλλαγή μεταδεδομένων. Οι κύριες επιλογές είναι οι εξής:

* **-atime**: Αναζήτηση βάσει της τελευταίας πρόσβασης στο αρχείο.
* **-mtime**: Αναζήτηση βάσει της τελευταίας τροποποίησης του αρχείου.
* **-ctime**: Αναζήτηση βάσει της τελευταίας τροποποίησης των μεταδεδομένων του αρχείου (όπως δικαιώματα ή ιδιοκτησία).

Κάθε μία από αυτές τις επιλογές δέχεται έναν ακέραιο αριθμό που καθορίζει την περίοδο αναζήτησης:

|  |  |
| --- | --- |
| **Σύμβολο** | **Ερμηνεία** |
| +n | Αρχεία που τροποποιήθηκαν πριν από n ημέρες και παλαιότερα. |
| -n | Αρχεία που τροποποιήθηκαν τις τελευταίες n ημέρες. |
| n | Αρχεία που τροποποιήθηκαν ακριβώς πριν από n ημέρες. |

Παραδείγματα αναζήτησης με ημερομηνία

Αρχεία που τροποποιήθηκαν τις τελευταίες 3 ημέρες στον home κατάλογο:

[user@host ~]$ find ~ -mtime -3

Αρχεία στα οποία είχαμε πρόσβαση ακριβώς πριν από 3 ημέρες στον κατάλογο /tmp:

[user@host ~]$ find /tmp -atime 3

Αρχεία που τροποποιήθηκαν τα τελευταία 30 λεπτά:

[user@host ~]$ find /var/log -mmin -30

**Αναζήτηση με βάση το μέγεθος**

Η επιλογή -size μας επιτρέπει να φιλτράρουμε αρχεία σύμφωνα με το μέγεθός τους. Το μέγεθος εκφράζεται με αριθμούς και μπορεί να συνοδεύεται από έναν δείκτη μονάδας:

|  |  |
| --- | --- |
| **Μονάδα** | **Περιγραφή** |
| c | Bytes |
| K | KiloBytes (1024 Bytes) |
| M | MegaBytes (1024 KB) |
| G | GigaBytes (1024 MB) |

Επίσης μπορούν να χρησιμοποιηθούν τα σύμβολα + και – με το ίδιο νόημα με την αναζήτηση με βάση την Ημερομηνία

Παραδείγματα αναζήτησης

Αρχεία μεγαλύτερα από 100 Megabytes στο σύστημα:

[user@host ~]$ sudo find / -size +100M

/boot/initramfs-0-rescue-1bc622f64dc74baabd140f5d216f8b05.img

/proc/kcore

...

Αρχεία μικρότερα από 1 KiloByte στον κατάλογο /tmp:

[user@host ~]$ find /tmp -size -1K

Αρχεία ακριβώς 512 Bytes στον κατάλογο /var/spool:

[user@host ~]$ find /var/spool -size 512c

**Αναζήτηση με βάση τα δικαιώματα**

Η εντολή find μας δίνει τη δυνατότητα να αναζητήσουμε αρχεία και καταλόγους με βάση τα δικαιώματά τους. Παρότι μπορεί να φαίνεται περίπλοκο, η σύνταξη είναι ισχυρή και ευέλικτη, επιτρέποντάς μας να περιγράψουμε διαφορετικούς τύπους αναζήτησης.

Η σύνταξη του κριτηρίου είναι η εξής:

-perm <τύπος> <δικαιώματα>

Τα δικαιώματα μπορούν να δηλωθούν είτε με την αριθμητική τιμή (π.χ., 0602) είτε με τη συμβολική μορφή (π.χ., u=rw,o=w). Ο τύπος είναι προαιρετικός και καθορίζει το πώς η find θα συγκρίνει τα δικαιώματα. Οι διαθέσιμοι τύποι είναι οι εξής:

**Κενό** (προεπιλογή): Εντοπίζει αρχεία που έχουν ακριβώς τα συγκεκριμένα δικαιώματα.

[user@host ~]$ find ~ -perm 0020

Η παραπάνω εντολή βρίσκει αρχεία με ακριβώς δικαιώματα g=w.

**Παύλα (-)**: Εντοπίζει αρχεία που έχουν τουλάχιστον τα συγκεκριμένα δικαιώματα. Αυτή είναι η πιο συχνή περίπτωση.

[user@host ~]$ find ~ -perm -u=x

Βρίσκει αρχεία στον home κατάλογο όπου ο ιδιοκτήτης έχει δικαίωμα εκτέλεσης.

**Κάθετος (/):** Εντοπίζει αρχεία που έχουν οποιοδήποτε από τα δικαιώματα.

[user@host ~]$ find /usr/bin -perm /u=w,g=x

Η παραπάνω εντολή εντοπίζει αρχεία στον κατάλογο /usr/bin όπου είτε ο ιδιοκτήτης έχει δικαίωμα εγγραφής είτε η ομάδα ιδιοκτήτη έχει δικαίωμα εκτέλεσης.

**Χρήση της ενέργειας exec στην find**

Η επιλογή -exec της εντολής find είναι εξαιρετικά χρήσιμη, καθώς μας επιτρέπει να εκτελέσουμε ενέργειες στα αρχεία που εντοπίζει η find, αντί να περιοριζόμαστε μόνο στην εμφάνιση των αποτελεσμάτων. Με αυτόν τον τρόπο, μπορούμε να αυτοματοποιήσουμε σύνθετες εργασίες απευθείας από τη γραμμή εντολών.

Η βασική σύνταξη για τη χρήση της επιλογής -exec είναι:

find <κατάλογος> <κριτήρια> -exec <εντολή> {} \;

* {}: Αντικαθίσταται από το όνομα κάθε αρχείου που ταιριάζει στα κριτήρια.
* \;: Υποδεικνύει το τέλος της εντολής που θέλουμε να εκτελεστεί.

Έστω ότι θέλουμε να υπολογίσουμε πόσες γραμμές περιέχει κάθε αρχείο .txt στον home κατάλογό μας. Αυτό μπορούμε να το επιτύχουμε εφαρμόζοντας την εντολή wc -l στα αρχεία που βρίσκει η find:

[user@host ~]$ find ~ -name "\*.txt" -exec wc -l {} \;

**Χρήση πολλαπλών κριτηρίων**

Όταν συνδυάζουμε περισσότερα από ένα κριτήρια στην find, η προεπιλεγμένη συμπεριφορά είναι να απαιτείται η ταυτόχρονη ικανοποίηση όλων των κριτηρίων (λογικό ΚΑΙ). Για παράδειγμα:

[user@host ~]$ find ~ -name "\*.sh" -atime -20 2>/dev/null

Η παραπάνω εντολή εμφανίζει όλα τα αρχεία στον home κατάλογο που:

* Το όνομά τους έχει την κατάληξη .sh.
* Έχουν προσπελαστεί τις τελευταίες 20 ημέρες.

Αν θέλουμε να εμφανίσουμε αρχεία που ικανοποιούν είτε το ένα κριτήριο είτε το άλλο (λογικό Ή), χρησιμοποιούμε τον τελεστή -o:

[user@host ~]$ find ~ -name "\*.sh" -o -atime -20 2>/dev/null

**Περιορισμός βάθους αναζήτησης**

Η find αναζητά αρχεία στον κατάλογο που ορίζουμε, καθώς και σε όλους τους υποκαταλόγους του. Για να περιορίσουμε το βάθος της αναζήτησης, χρησιμοποιούμε την επιλογή -maxdepth. Για παράδειγμα:

[user@host ~]$ find ~ -maxdepth 1 -name "\*.sh"

Η find ψάχνει μόνο στον home κατάλογο (βάθος 1) για αρχεία με κατάληξη .sh.

## Άσκηση: Ρύθμιση της πρόσβασης στο σύστημα αρχείων

### Εργασία 1: Λειτουργίες διαχείρισης Συσκευών αποθήκευσης

Στόχος της εργασίας αυτής είναι η πρακτική κατανόηση διαφόρων λειτουργιών διαχείρισης που σχετίζονται με τις συσκευές αποθήκευσης στο Linux. Μέσα από μια σειρά σεναρίων, θα εξερευνήσουμε βασικές λειτουργίες όπως η προβολή πληροφοριών, η δημιουργία partitions, η μορφοποίηση και η αυτόματη προσάρτηση συστημάτων αρχείων.

**Σενάριο 1: Προβολή πληροφοριών σχετικά με τις συσκευές αποθήκευσης**

Στόχος: Θέλουμε να εμφανίσουμε πληροφορίες σχετικά με τις συσκευές αποθήκευσης που υπάρχουν στο σύστημά μας, ώστε να κατανοήσουμε τη διάταξη και τη χρήση τους.

Για τον σκοπό αυτό, χρησιμοποιούμε την εντολή lsblk, η οποία παρέχει μια ιεραρχική προβολή των συσκευών αποθήκευσης και των partitions τους.

[user@host ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

sr0 11:0 1 1024M 0 rom

nvme0n1 259:0 0 20G 0 disk

├─nvme0n1p1 259:1 0 1G 0 part /boot

└─nvme0n1p2 259:2 0 19G 0 part

├─rl\_localhost--live-root 253:0 0 17G 0 lvm /

└─rl\_localhost--live-swap 253:1 0 2G 0 lvm [SWAP]

[user@host ~]$

Στα αποτελέσματα της εντολής παρατηρούμε ότι το σύστημα μας διαθέτει έναν σκληρό δίσκο nvme0n1 με δύο partitions (nvme0n1p1 και nvme0n1p2). Το πρώτο partition είναι προσαρτημένο στον κατάλογο /boot, ενώ το δεύτερο ανήκει σε ένα λογικό σύστημα LVM που περιλαμβάνει δύο λογικά volumes (root και swap). Να τονίσουμε ότι η έξοδος της lsblk εξαρτάται από τη σύνθεση του μηχανήματός μας.

**Σενάριο 2: Δημιουργία partitions με GPT**

Στόχος: Θέλουμε να δημιουργήσουμε δύο partitions σε έναν νέο δίσκο, χρησιμοποιώντας το GPT ως τύπο πίνακα διαμερίσματος. Για την πραγματοποίηση αυτού του σεναρίου απαιτείται η προσθήκη μιας καινούργιας συσκευής αποθήκευσης στο σύστημά μας. Η διαδικασία αυτή γίνεται είτε με τη φυσική σύνδεση της συσκευής σε ένα πραγματικό σύστημα είτε με την προσθήκη μιας εικονικής συσκευής σε μία εικονική μηχανή. Στο παράδειγμα μας η συσκευή αυτή είναι 15GB και έχει αναγνωριστεί από την υπηρεσία udev με το όνομα /dev/sda.

[user@host ~]$ lsblk | head -1; lsblk | grep sd

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

sda 8:0 0 15G 0 disk

Θέλουμε να χωρίσουμε την συσκευή αυτή σε δύο partition, το πρώτο partition θα καταλαμβάνει 9GB και το δεύτερο 6GB. Χρησιμοποιούμε την εντολή parted για τη διαχείριση του δίσκου:

[user@host ~]$ sudo parted /dev/sda

GNU Parted 3.5

Using /dev/sda

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) mklabel gpt

(parted) mkpart ext4 0GB 9GB

(parted) mkpart ext4 9GB 15GB

(parted) p

Model: ATA VMware Virtual S (scsi)

Disk /dev/sda: 16.1GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Disk Flags:

Number Start End Size File system Name Flags

1 1049kB 9000MB 8999MB ext4

2 9000MB 15.0GB 6000MB ext4

Στο παραπάνω παράδειγμα παρατηρούμε:

* Χρησιμοποιούμε το mklabel gpt για να επιλέξουμε το GPT ως τύπο πίνακα διαμερίσματος.
* Η mkpart μας επιτρέπει να ορίσουμε το εύρος κάθε partition.
* Με την p (print) επαληθεύουμε τη δημιουργία των partitions.

Σημείωση: Σε αντίθεση με την fdisk, η parted δημιουργεί άμεσα τα partitions χωρίς την ανάγκη αποθήκευσης.

**Σενάριο 3: Μορφοποίηση των partitions με σύστημα αρχείων**

Στόχος: Θέλουμε να μορφοποιήσουμε ένα partition, εφαρμόζοντας σε αυτό ένα σύστημα αρχείων κατάλληλο για την επιθυμητή χρήση του.

Προτού χρησιμοποιήσουμε ένα partition, πρέπει να του εφαρμόσουμε ένα σύστημα αρχείων. Για να δούμε τις διαθέσιμες επιλογές, εκτελούμε:

[user@host ~]$ ls /usr/sbin/mkfs\*

/usr/sbin/mkfs /usr/sbin/mkfs.ext4 /usr/sbin/mkfs.vfat

/usr/sbin/mkfs.cramfs /usr/sbin/mkfs.fat /usr/sbin/mkfs.xfs

/usr/sbin/mkfs.ext2 /usr/sbin/mkfs.minix

/usr/sbin/mkfs.ext3 /usr/sbin/mkfs.msdos

Για να μορφοποιήσουμε το πρώτο partition (/dev/sda1) που δημιουργήσαμε στο προηγούμενο σενάριο με το σύστημα αρχείων ext4:

[user@host ~]$ sudo mkfs.ext4 /dev/sda1

mke2fs 1.46.5 (30-Dec-2021)

Creating filesystem with 2196992 4k blocks and 549440 inodes

Filesystem UUID: 356f3eb3-3521-47bd-a7a4-aeda38ae9229

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done

Writing inode tables: done

Creating journal (16384 blocks): done

Writing superblocks and filesystem accounting information: done

[user@host ~]$

**Σενάριο 4: Αυτόματα προσάρτηση συστήματος αρχείων**

Στόχος: Θέλουμε να ρυθμίσουμε το σύστημα αρχείων που δημιουργήσαμε σε προηγούμενο σενάριο να προσαρτάται αυτόματα στον κατάλογο /mnt/custom\_log.

Δημιουργία καταλόγου στον οποίο θα γίνει η προσάρτηση

[user@host ~]$ sudo mkdir -p /mnt/custom\_log

Ενημέρωση του καταλόγου /etc/fstab με την ακόλουθη εγγραφή

/dev/sda1 /mnt/custom\_log ext4 defaults 0 0

Ενημέρωση του συστήματος για τις αλλαγές

[user@host ~]$ sudo mount -a

mount: (hint) your fstab has been modified, but systemd still uses

the old version; use 'systemctl daemon-reload' to reload.

[user@host ~]$ sudo systemctl daemon-reload

Επαλήθευση της δημιουργίας της προσάρτησης με την χρήση της εντολής df -h.

### Εργασία 2: Λειτουργίες του LVM

Στόχος: Θέλουμε να εμφανίσουμε λεπτομερείς πληροφορίες σχετικά με τα Physical Volumes (PVs), τα Volume Groups (VGs) και τα Logical volumes (LVs) που υπάρχουν στο σύστημά μας, ώστε να κατανοήσουμε τη δομή και τη χρήση τους.

Για την προβολή αυτών των πληροφοριών χρησιμοποιούμε τις εντολές pvs, vgs και lvs, όπως φαίνεται παρακάτω:

[user@host ~]$ sudo pvs

PV VG Fmt Attr PSize PFree

/dev/nvme0n1p2 rl\_localhost-live lvm2 a-- <19.00g 0

[user@host ~]$ sudo vgs

VG #PV #LV #SN Attr VSize VFree

rl\_localhost-live 1 2 0 wz--n- <19.00g 0

[user@host ~]$ sudo lvs

LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert

root rl\_localhost-live -wi-ao---- <17.00g

swap rl\_localhost-live -wi-ao---- 2.00g

Ανάλυση:

* Η εντολή pvs εμφανίζει τα physical volumes, αναφέροντας το μέγεθός τους και το αν είναι διαθέσιμα (PFree).
* Η εντολή vgs εμφανίζει τα volume groups, την πλήρη χωρητικότητα (VSize), και το διαθέσιμο μέγεθος (VFree).
* Η εντολή lvs εμφανίζει τα logical volumes, μαζί με πληροφορίες για το μέγεθός τους και τη χρήση τους.

Από τα παραπάνω καταλαβαίνουμε ότι:

* Υπάρχει ένα volume group με το όνομα rl\_localhost-live, στο οποίο έχει ανατεθεί το physical volume /dev/nvme0n1p2.
* Από το volume group έχουν δημιουργηθεί δύο logical volumes: το root (17GB) και το swap (2GB).

Επιπλέον, για να δούμε πόσο γεμάτα είναι τα συστήματα αρχείων, χρησιμοποιούμε την εντολή df -h:

[user@host ~]$ df -h

Filesystem Size Used Avail Use% Mounted on

devtmpfs 4.0M 0 4.0M 0% /dev

tmpfs 1.8G 0 1.8G 0% /dev/shm

tmpfs 726M 9.7M 716M 2% /run

/dev/mapper/rl\_localhost--live-root 17G 6.6G 11G 39% /

tmpfs 1.8G 16K 1.8G 1% /tmp

Η παραπάνω εντολή δείχνει ότι το σύστημα αρχείων που είναι προσαρτημένο στο root (/) χρησιμοποιεί το 39% του διαθέσιμου χώρου του.

**Σενάριο 2: Δημιουργία νέου Logical Volume**

Στόχος: Θέλουμε να δημιουργήσουμε ένα νέο volume group με το όνομα log\_srvc, να του αναθέσουμε ένα φυσικό volume, και να δημιουργήσουμε ένα νέο logical volume σε αυτό.

Βήμα 1: Δημιουργία Physical Volume

Αρχικά, χρησιμοποιούμε το partition /dev/sda2 που δημιουργήσαμε σε σενάριο προηγούμενης εργασίας για να δημιουργήσουμε ένα νέο Physical Volume:

[user@host ~]$ sudo pvcreate /dev/sda2

Physical volume "/dev/sda2" successfully created.

[user@host ~]$ sudo pvs

PV VG Fmt Attr PSize PFree

/dev/nvme0n1p2 rl\_localhost-live lvm2 a-- <19.00g 0

/dev/sda2 lvm2 --- <5.59g <5.59g

Βήμα 2: Δημιουργία Volume Group

Στη συνέχεια, δημιουργούμε το volume group log\_srvc και του αναθέτουμε το νέο physical volume:

[user@host ~]$ sudo vgcreate log\_srvc /dev/sda2

Volume group "log\_srvc" successfully created

[user@host ~]$ sudo vgs

VG #PV #LV #SN Attr VSize VFree

log\_srvc 1 0 0 wz--n- <5.59g <5.59g

rl\_localhost-live 1 2 0 wz--n- <19.00g 0

Το volume group log\_srvc δημιουργήθηκε με συνολική χωρητικότητα 5.59GB.

Βήμα 3: Δημιουργία Logical Volume

Δημιουργούμε ένα logical volume που χρησιμοποιεί όλο τον ελεύθερο χώρο του volume group:

user@host ~]$ sudo lvcreate -l 100%FREE log\_srvc -n log\_data

Logical volume "log\_data" created.

[user@host ~]$ sudo lvs

LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert

log\_data log\_srvc -wi-a----- <5.59g

root rl\_localhost-live -wi-ao---- <17.00g

swap rl\_localhost-live -wi-ao---- 2.00g

Βήμα 4: Μορφοποίηση και προσάρτηση

Για να χρησιμοποιήσουμε το logical volume, το μορφοποιούμε με το σύστημα αρχείων ext4 και το προσαρτούμε σε έναν κατάλογο:

[user@host ~]$ sudo mkfs.ext4 /dev/log\_srvc/log\_data

…

[user@host ~]$ sudo mkdir -p /mnt/log\_data

[user@host ~]$ sudo mount /dev/log\_srvc/log\_data /mnt/log\_data

### Εργασία 3: Αναζήτηση αρχείων και καταλόγων με την χρήση των εντολών find και locate

Σε αυτή την άσκηση, θα εξερευνήσουμε τη χρήση της εντολής find, εφαρμόζοντας διάφορα σενάρια για την αναζήτηση αρχείων και καταλόγων. Τα σενάρια καλύπτουν πολλαπλά κριτήρια αναζήτησης, τα οποία μπορούν να προσαρμοστούν για διάφορες ανάγκες.

**Σενάριο 1: Αναζήτηση με την εντολή find**

Στόχος: Εξοικείωση με τα κριτήρια αναζήτησης που παρέχει η εντολή find.

**Εύρεση αρχείου με συγκεκριμένο όνομα**

Θέλουμε να βρούμε το αρχείο με τίτλο Budget\_2024.pdf που βρίσκεται στον κατάλογο /home/account:

find /home/account -name "Budget\_2024.pdf"

**Εύρεση μεγάλων αρχείων**

Αναζητούμε αρχεία με μέγεθος μεγαλύτερο από 100 MB σε ολόκληρο το σύστημα:

find / -size +100M

**Αναζήτηση βάσει ημερομηνίας τροποποίησης**

Θέλουμε να βρούμε όλα τα αρχεία στον κατάλογο /var/reports που έχουν τροποποιηθεί τις τελευταίες 7 ημέρες:

find /var/reports -mtime -7

**Εύρεση καταλόγων**

Εντοπίζουμε όλους τους υποκαταλόγους στον κατάλογο /home/user/projects:

find /home/user/projects -type d

**Αναζήτηση βάσει ιδιοκτήτη**

Βρίσκουμε όλα τα αρχεία στον κατάλογο /tmp που ανήκουν στον χρήστη aweb:

find /tmp -user aweb

**Εύρεση αρχείων με συγκεκριμένα δικαιώματα**

Εντοπίζουμε αρχεία στον κατάλογο /home/user/documents που έχουν τουλάχιστον το δικαίωμα εκτέλεσης για την ομάδα:

find /home/user/documents -type f -perm -010

**Αρχεία που δεν είναι readable από τον ιδιοκτήτη**

Αναζητούμε αρχεία στον κατάλογο /home/user/documents που δεν έχουν τουλάχιστον δικαίωμα ανάγνωσης από τον ιδιοκτήτη τους:

find /home/user/documents -type f ! -perm -400

**Συνδυασμένα κριτήρια (ΚΑΙ)**

Θέλουμε να βρούμε όλα τα αρχεία στον κατάλογο /etc που έχουν τουλάχιστον δικαίωμα εγγραφής για την ομάδα ιδιοκτήτη και ανήκουν στον χρήστη root.

find /etc -type f -perm -020 -user root

**Συνδυασμένα κριτήρια (Η)**

Βρίσκουμε όλα τα αρχεία σε έναν συγκεκριμένο κατάλογο που είτε ανήκουν στον χρήστη user1 είτε ανήκουν στην ομάδα group1:

find /path/to/search -type f \( -user user1 -o -group group1 \)

Σενάριο 2: Αναζήτηση με την εντολή locate

**Εύρεση αρχείου με συγκεκριμένο όνομα**

Θέλουμε να βρούμε το αρχείο με όνομα example.conf στον δίσκο μας.

locate example.conf

**Εύρεση αρχείων με την χρήση κανονικών εκφράσεων**

Θέλουμε να βρούμε όλα τα αρχεία που έχουν μέσα τους την λέξη Project1, Project2 ή Project3.

locate -r ‘Project[1-3].txt’

**Εύρεση Όλων των Αρχείων με Κατάληξη .log**

Θέλουμε να βρούμε όλα τα αρχεία στον υπολογιστή μας που έχουν κατάληξη .log.

locate \*.log

# ΒΙΒΛΙΟΓΡΑΦΙΑ

* Banâtre, M., & Jouvelot, P. (1990). Process Management in Unix/Linux. SpringerLink.
* Bar, M. (2006). Linux File Systems. Sams Publishing.
* Barrett, D. J. (2016). Linux Pocket Guide: Essential Commands. O'Reilly Media.
* Biriukov, D. (2021). Terminals and Pseudoterminals. Retrieved from https://biriukov.dev/docs/fd-pipe-session-terminal/4-terminals-and-pseudoterminals/.
* Boolean World. (n.d.). A Guide to the Linux 'Find' Command. Retrieved from https://www.booleanworld.com/guide-linux-find-command/.
* Clausen, A., Kreuter, R. M., & Polzer, L. P. (2019). GNU Parted User Manual. Free Software Foundation.
* Dasgupta, A. (2023). Understanding Software Package Management in Linux. Linux Journal. Retrieved from https://www.linuxjournal.com.
* DigitalOcean. (2022). An Introduction to Storage Terminology and Concepts in Linux. Retrieved from https://www.digitalocean.com/community/tutorials/an-introduction-to-storage-terminology-and-concepts-in-linux.
* Donovan, J. S. (2015). The Linux Process Manager: The Internals of Scheduling, Interrupts, and Signals. Amazon.
* FOSS Linux. (2023). The ABCs of Linux Signals: SIGINT, SIGTERM, and SIGKILL Explained. Retrieved from https://www.fosslinux.com/121761/the-abcs-of-linux-signals-sigint-sigterm-and-sigkill-explained.htm.
* GeeksforGeeks. (n.d.). locate command in Linux with Examples. Retrieved from https://www.geeksforgeeks.org/locate-command-in-linux-with-examples/.
* International Journal of Trend in Research and Development. (2020). An Analysis of Linux Operating System. IJTRD.
* Kerrisk, M. (2010). The Linux Programming Interface. No Starch Press.
* Linux Foundation. (2023). Introduction to Linux Package Management. Retrieved from https://training.linuxfoundation.org.
* Linux manual page. ps. Retrieved from https://man7.org/linux/man-pages/.
* Linux manual page. setpgid. Retrieved from https://man7.org/linux/man-pages/.
* Linux manual page. top. Retrieved from https://man7.org/linux/man-pages/man1/top.1.html.
* Linux manual page. tty. Retrieved from https://man7.org/linux/man-pages/.
* LinuxOrgKillSignals. (n.d.). Kill Commands and Signals. Retrieved from https://www.linux.org/threads/kill-commands-and-signals.8881/.
* Lonston, B. (2018). Create and Mount Filesystems in Linux. Retrieved from https://www.linuxsysadmins.com/create-and-mount-filesystems-in-linux/.
* Love, R. (2013). Linux System Programming: Talking Directly to the Kernel and C Library. O'Reilly Media.
* Negus, C. (2020). The Linux Bible. Wiley.
* Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). UNIX and Linux System Administration Handbook. Addison-Wesley Professional.
* PhoenixNAP. (2023). Linux locate Command: Syntax, Options, Examples. Retrieved from https://phoenixnap.com/kb/locate-command-in-linux.
* Red Hat. (n.d.). Killing Zombies Linux Style. Retrieved from https://www.redhat.com/sysadmin/killing-zombies-linux-style.
* Red Hat. (n.d.). The LVM Logical Volume Manager. Retrieved from https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/4/html/cluster\_logical\_volume\_manager/lvm\_overview.
* Red Hat, Inc. (2024). DNF Command Reference Guide. Retrieved from https://dnf.readthedocs.io.
* Red Hat Enterprise Linux Documentation. Retrieved from https://www.redhat.com.
* Rocky Linux Repositories. Retrieved from https://rockylinux.pkgs.org/.
* Salter, J. (2018). Understanding Linux Filesystems: ext4 and Beyond. Retrieved from https://opensource.com/article/18/4/ext4-filesystem.
* Smith, K. (2021). Mastering Linux Administration. Packt Publishing.
* Smith, R. (2023). Mastering RPM and YUM for Red Hat Systems. O'Reilly Media.
* Sobell, M. G. (2018). A Practical Guide to Linux Commands, Editors, and Shell Programming. Prentice Hall.
* Soete, M. (2023). Package Management Systems: An Overview of APT, RPM, and DNF. Opensource.com. Retrieved from https://opensource.com.
* Stevens, W. R. (2013). Advanced Programming in the UNIX Environment. Addison-Wesley Professional.
* Tecmint. (n.d.). 35 Practical Examples of Linux Find Command. Retrieved from https://www.tecmint.com/35-practical-examples-of-linux-find-command/.
* TekNeed. (2020). Storage Management in Linux Explained with Examples. Retrieved from https://tekneed.com/storage-management-in-linux-explained-with-examples/.
* The Linux Documentation Project. Retrieved from https://tldp.org.
* Torvalds, L., & Diamond, D. (2001). Just for Fun: The Story of an Accidental Revolutionary. HarperBusiness.
* Ubuntu Documentation Team. (2024). APT User Guide. Retrieved from https://help.ubuntu.com.
* Van Vugt, S. (2015). Red Hat RHCSA/RHCE 7 Cert Guide: Red Hat Enterprise Linux 7 (EX200 and EX300). Pearson IT Certification.