



# Curs 1

## Introducere

---

Gestiunea serviciilor de rețea (GSR)  
6 octombrie 2016

Departamentul de Calculatoare, Comunitatea RLUG

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări

- ▶ Pentru oameni tehnici (nu neapărat administratori)
- ▶ Înțelegerea funcționării sistemelor
- ▶ Servicii de rețea, protocoale, interoperare
- ▶ Linux

- ▶ Dobândirea și îmbunătățirea competențelor de troubleshooting la nivelul sistemelor de calcul
- ▶ Obținerea de cunoștințe de construire și mentenanță a unei infrastructuri IT de mici dimensiuni (small setup)
- ▶ Îmbunătățirea abilităților de înțelegere și descriere a cerințelor și soluțiilor IT în ansamblul acestora, înțelegerea avantajelor și dezavantajelor unei soluții într-un context dat

- ▶ Colaborare între Departamentul de Calculatoare și Comunitatea RLUG
- ▶ Ocazia de a interacționa cu oameni din industrie care au o experiență bogată în utilizarea și configurarea sistemelor Linux
- ▶ Mai multe detalii pe pagina echipei de pe OCW:  
<http://ocw.cs.pub.ro/courses/gsr>

- ▶ Sala de laborator: PR706; Ubuntu GNOME 16.04
- ▶ cs.curs.pub.ro:  
`http://cs.curs.pub.ro/2016/course/view.php?id=197`
- ▶ wiki OCW: `http://ocw.cs.pub.ro/courses/gsr`

- ▶ laborator: 2 puncte
- ▶ test practic: 2.5 puncte (în sesiune)
- ▶ teste de curs: 2 puncte (5 teste, la cursurile impare, acoperă materia din cele mai recente două cursuri)
- ▶ test grilă final: 2 puncte (44 de întrebări, în sesiune)
- ▶ proiect: 2.5 puncte (evaluat în săptămâna a 8-a și în sesiune)

- ▶ Curs 1: 6 octombrie 2016, în intervalul 18:00-20:00, sala PR106
- ▶ Laboratorul 1: 13 octombrie 2016, în intervalele 16:00-18:00 și 20:00-22:00, sala PR706
- ▶ Teste de curs înaintea cursurilor 3, 5, 7, 9 și 11: durează 10 minute la începutul cursului
- ▶ Împartirea pe cele două intervale pentru laborator se va face pe platforma moodle, miercuri 12 octombrie. Urmăriți mesajele de pe forum.



1. Introducere și procese
2. Servicii sistem și server basics
3. Serviciile de DHCP și SSH
4. Serviciul de DNS
5. Serviciul de e-mail
6. Servicii de file sharing
7. Servicii web
8. Securitate: PKI și X.509
9. Securitate: SSL și TLS
10. Securitate: hardening a server
11. Acces la distanță
12. After GSR

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări

- ▶ Un programator nu trebuie să fie cunoscător doar al limbajului
- ▶ Cunoașterea componentelor din sistemul folosit: sisteme de calcul, protocoale, utilizatori

- ▶ No proactiveness
- ▶ Rigid software development
- ▶ Impossible to cover all corner cases

- ▶ Servicii uzuale
- ▶ Protocoale asociate serviciilor
- ▶ Modul în care se abordează probleme/cerințe legate de servicii și protocoale

- ▶ Povestioare cu nevoi ale persoanelor și organizațiilor
- ▶ Soluții posibile pentru rezolvarea nevoilor
- ▶ Motivele alegerii unei soluții sau a alteia
- ▶ Întâi conceptual („pădurea”: componente, protocoale)...
- ▶ ...apoi implementările particulare („copacii”: soluții/servicii)
- ▶ Diferențe între implementările particulare

- ▶ Crearea unei infrastructuri de dezvoltare folosind AWS
  - ▶ User authentication
  - ▶ Code repositories
  - ▶ Build servers
  - ▶ Publishing servers
  - ▶ Automated QA for functionality and performance
  - ▶ Detailed reporting
  - ▶ Detailed monitoring
  - ▶ Security

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări



- ▶ Rețelistică (networking): adresa IP (protocol, adresare, subnetting, rutare, comenzi Linux), TCP (protocol, porturi, servicii, netstat), DNS (protocol, funcționare, clienți DNS, troubleshooting), firewalling. Sunt descrise în laboratoarele cursului de Rețele Locale (RL).
- ▶ Utilizare Linux de bază: sistemul de fișiere (concepte, tipuri de fișiere, comenzi, permisiuni), procese (concepte, comenzi, monitorizare, investigare), editare (Vim de bază, căutare de informații), documentare (pagini de manual, căutare în pagini de manual). Sunt descrise în laboratoarele cursului de Utilizarea Sistemelor de Operare (USO).
- ▶ Shell scripting de bază: redirectări, pipe-uri, expandare de comenzi, one linere, if, for, while. Sunt descrise în laboratoarele cursului de Utilizarea Sistemelor de Operare (USO).

- ▶ **Utilizarea Sistemelor de Operare**
- ▶ **Rețele de Calculatoare**
- ▶ **Sisteme de Operare**

- ▶ `http://ocw.cs.pub.ro/courses/uso/`
- ▶ `http://ocw.cs.pub.ro/courses/rl/`

Aspecte administrative

Servicii de rețea

Prerequisites

**FHS: File System Hierarchy Standard**

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări

---

Intrare	Rol
/	directorul rădăcină
/bin	comenzi binare (executabile) importante
/home	conține directoarele de bază ale utilizatorilor
/usr	sistem de fișiere secundar (/usr/bin, /usr/lib)
/var	date variabile (de obicei fișiere jurnal sau cache-uri)
/etc	fișiere de configurare
/boot	bootloader și informații de boot
/lib	biblioteci utilizate de sistem
/dev	intrări în sistemul de fișiere pentru dispozitive hardware
/root	home-ul superuser-ului (root)
~	scurtătură shell pentru directorul home al utilizatorului curent
~student	scurtătură shell pentru directorul home al utilizatorului student

---

- ▶ Standard creat de Linux Foundation
- ▶ Folosit de toate distribuțiile Linux
- ▶ Distribuțiile pot extinde FHS (ex. /sys).

- ▶ În Linux totul este un fișier (**/etc/inittab**, **/dev/sd\***, **/dev/mem**)
- ▶ În funcție de tipul acestuia (fișier, director, pipe, block, char) se decide modalitatea de acces

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări



- ▶ Adăugare: **useradd**
- ▶ Ștergere: **userdel** (-r dacă vrem să ștergem și \$HOME)

- ▶ În acest fișier sunt ținute informațiile despre utilizatorii existenți pe sistem: utilizator, UID, GID, shell, \$HOME
- ▶ **NU** editați de mână acest fișier. Folosiți utilitarele puse la dispoziție de sistem: **chsh**, **chgrp**, **chage**, **chfn**, **passwd**, **adduser**, **userdel**
- ▶ **vipw** permite editarea oarecum sigură a acestui fișier (pentru utilizatori avansați, folosiți cu mare grijă)

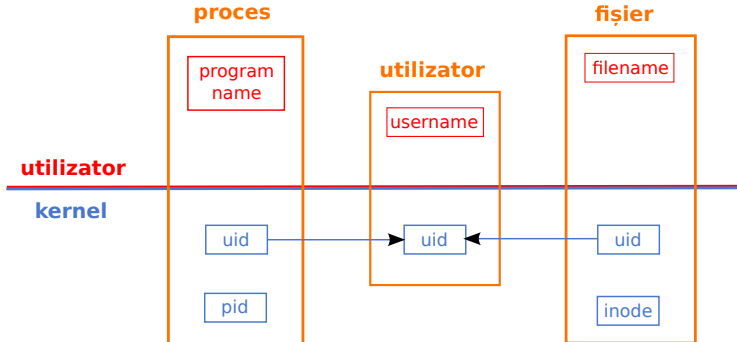
- ▶ În acest fișier sunt ținute hash-urile parolelor utilizatorilor, data expirării contului, dacă utilizatorul este dezactivat etc.
- ▶ La fel ca în cazul /etc/passwd, nu este recomandată editarea manuală a acestui fișier.

- ▶ Filosofia \*NIX spune că un program trebuie să facă un singur lucru și să-l facă bine
- ▶ Pe un sistem Linux avem mai multe conturi pentru a putea separa în mod facil rularea și controlul aplicațiilor (serverul web ruleaza sub un utilizator dedicat, la fel și serverul de DNS)

## Folosirea ls pentru a afla metadatele

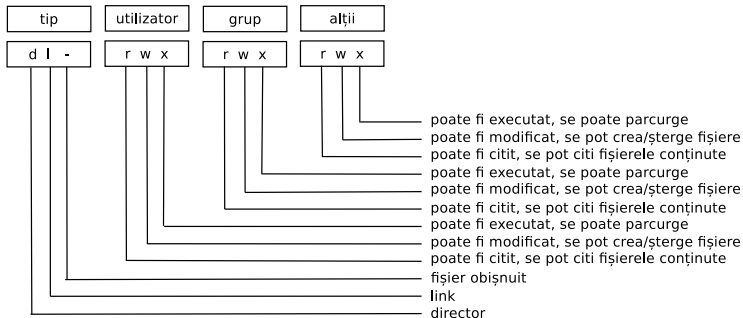
```
$ ls -li  
261222 -rw-rw-r-- 1 sin sin    0 Oct  8 15:15 a  
266083 drwxrwxr-x 2 sin sin 4096 Oct  8 15:15 b  
266084 prw-rw-r-- 1 sin sin    0 Oct  8 15:15 c
```

- ▶ inode-ul unde este "stocat" fișierul
- ▶ tipul de fișier (fișier, director, pipe, block device, character device)
- ▶ ce permisiuni are utilizatorul ce deține fișierul
- ▶ ce permisiuni are grupul ce deține fișierul
- ▶ ce permisiuni au "others"
- ▶ utilizatorul care deține fișierul
- ▶ grupul care deține fișierul
- ▶ dimensiunea fișierului
- ▶ data ultimei modificări
  
- ▶ **chmod** - utilitar pentru schimbarea permisiunilor pe fișiere
- ▶ **ls** - utilitar cu care se pot vedea permisiunile pe fișiere



- ▶ **chmod** - utilitar pentru schimbarea permisiuni. Se pot modifica permisiunile folosind valori numerice în baza 8 sau valori textuale relative la utilizator, grup sau „others”
  - ▶ `chmod u+r,g-rw,o+rw nume_fisier`
  - ▶ `chmod 0406 nume_fisier`
- ▶ Permiuni speciale: `setuid`, `setgid`, „sticky” bit
  - ▶ `setuid`: fișierul este rulat cu permisiunile utilizatorului care este „owner” pe fișier
  - ▶ `setgid`: fișierul este rulat cu permisiunile grupului care este „owner” pe fișier
  - ▶ `sticky`: folosit pentru directoare. Fișierele din acel director pot fi șterse/redenumite doar de owner/root, independent de drepturile de write





- ▶ **root** este contul de utilizator care poate face orice\* pe un sistem Linux
- ▶ se folosește atunci când se fac instalări de pachete, se configurează servicii, se adaugă/șterg partiții, pe scurt se folosește când este nevoie de efectuarea unor acțiuni administrative
- ▶ Problema root: e ușor de cedat tentației de a îl folosi excesiv. Totul este mai ușor atunci când esti root, dar asta nu înseamnă că este și bine.

- ▶ **bad**: rulat totul ca root (login ca root, efectuarea tuturor operațiilor ca root)
- ▶ **good**: rulat ca root doar operațiile care au neapărată nevoie de permisiunile lui root folosind **sudo**, altfel totul se face sub un utilizator „neprivilegiat”

- ▶ Se folosește pentru programe care au nevoie să ruleze cu alte permisiuni decât cele ale utilizatorului curent
- ▶ Scenariu uzual: fișier executabil deținut de root, cu bitul de setuid setat și drept de execuție pentru alți utilizatori
- ▶ **ping** pe anumite sisteme are nevoie de `_setuid` ca `!root` pentru a funcționa cum trebuie (numai **root** poate crea un socket „raw”)

- ▶ **sudo** permite rularea unor comenzi ca și un alt utilizator (root sau altul)
- ▶ *with great power comes great responsibility* - nu abuzați de sudo pentru orice operațiune
- ▶ **/etc/sudoers** - conține informațiile despre mapări de tip **utilizator - comandă pe care o poate rula**
- ▶ **visudo** - permite editarea fișierului **/etc/sudoers** și verificarea că este totul în regulă ca sintaxă înainte de a-l salva

- ▶ Împlementează la nivel de Linux kernel un model de MAC (Mandatory Access Control)
- ▶ Limitează accesul la resurse (fișiere și rețea) pentru a preveni intruziunile și a limita efectele acestora
- ▶ Disponibil din v2.6 a Linux Kernel, și dezvoltat inițial de US NSA.

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări

- ▶ Kernelul (nucleul) este componenta care asigură „interfața” între aplicații și hardware
- ▶ Kernelul asigură de asemenea execuția aplicațiilor, accesul la memorie, limitarea resurselor etc.



- ▶ Un sistem modern trebuie să poată realiza mai multe „task”-uri în același timp
- ▶ Aceste „task”-uri sunt realizate prin procese

- ▶ O parte din procese sunt pornite direct de utilizator, altele rulează în background (daemoni/servicii)
- ▶ Procesele de sistem se împart în două categorii: thread-uri de kernel și daemons
  - ▶ thread-urile de kernel, deși apar în output-ul lui ps sau pstree, nu au PID și nu pot fi oprite
  - ▶ daemon-ii/aplicațiile de sistem sunt acele procese necesare pentru functionarea sistemului (ex. mingetty, udevd, login)

- ▶ **init** este „tata'lor” (PID 1)
- ▶ toate procesele sunt organizate ierarhic, pornind de la init
- ▶ în afară de procesele sistem importante, toate celelalte ruleaza sub utilizatori proprii (www-data, ftp, ssh etc.)
- ▶ fiecare utilizator are un numar predefinit de resurse pe care le poate folosi (descriptori de fișire, utilizare CPU, memorie etc.)

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări

- ▶ `./configure; make; make install`

- ▶ "Comerciale": RedHat, SuSe Linux
- ▶ Debian, CentOS, Ubuntu, ArchLinux, Mint etc.

- ▶ Nevoie diferite: unii au nevoie de stabilitate și suport îndelungat, alții vor ultimele versiuni de pachete și ultimele capacități (features)
- ▶ Pentru că se poate

- ▶ Pachetele sunt unitatea de lucru a distribuției
- ▶ Fiecare program este disponibil pe o distribuție de Linux sub forma de unul sau mai multe pachete în funcție de complexitatea acestuia



- ▶ `rpm -ql sudo` sau `dpkg -L sudo`

- ▶ O "dependență" a unui pachet reprezintă una sau mai multe componente software aflate în alte pachete fără de care acel pachet software nu poate funcționa corect.
- ▶ De exemplu **mysql-client** nu poate funcționa fără **mysql-libs**

- ▶ `yum remove emacs`
- ▶ `yum install vim`
  
- ▶ `dpkg --purge emacs`
- ▶ `apt-get install vim`

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări

- ▶ James Turnbull, Peter Lieverdink, Dennis Matotek – Pro Linux System Administration
- ▶ <http://ocw.cs.pub.ro/courses/gsr/>
- ▶ <http://debian.org/doc/user-manuals>
- ▶ <http://permissions-calculator.org>
- ▶ [https://en.wikipedia.org/wiki/Filesystem\\_Hierarchy\\_Standard](https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard)

Aspecte administrative

Servicii de rețea

Prerequisites

FHS: File System Hierarchy Standard

Utilizatori, permisiuni și posesie

Structura unui sistem de operare

Lucrul cu pachete

Sumar

Întrebări