Linux presentation

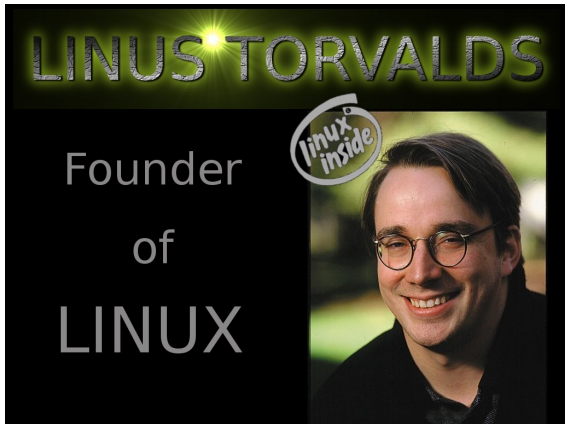
A BASIC INTRODUCTION

What is Linux ?

- Linux is a free and open source operating system.
- At it's core, the Linux operating system is derived from the Unix OS.
  - **Unix** was created in the 1960s by Dennis Ritchie and Ken Thompson, both of them **also invented the C programming language.**

Kenneth L. Thompson Dennis M. Ritchie

- Linux was initially named GNU and was developed by Richard Stallman
- **Linux** was the name of the kernel created in 1991 by Linux Torvalds, a student at the University of Helsinki.
- People started calling the GNU OS, Linux – because of the name of the kernel



What distributions of Linux do exist?

- Linux OS has multiple distributions (called distros) that are derived from it's initial deployment.
- Most of the are FREE and offer full functionality:
  - **Examples**:
    - Debian
    - Ubuntu
    - CentOS
    - OpenSUSE
    - Mint
    - Gentoo
    - Slackware.

What distributions of Linux do exist (II)?

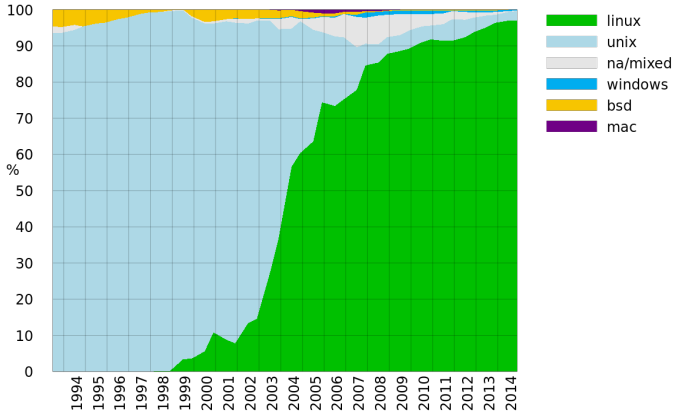
- Some examples of non-free (enterprise) Linux distros are:
  - **Red Hat** Enterprise Linux
  - **SUSE** Linux Enterprise Server
  - **Oracle** Linux
  - **Scientific** Linux
  - **Turbo** Linux
  - Linux **Mandriva**

  - For more info about various types of Linux:

  http://distrowatch.com/

Why should I care about Linux ?

- In September 2008 Steve Ballmer (Microsoft CEO) claimed 60% of servers run Linux and 40% run Windows Server. According to IDC's report covering Q2 2013, Linux was up to 23.2% of worldwide server revenue.
- Linux is used as:
  - Server (HTTP, FTP, DNS, file server, etc)
  - Desktop (it’s a free alternative to Microsoft’s Windows XP, Vista, 7, 8 family)
  - Supercomputer operating system:
    - According to Wikipedia & top500.org, over 95% of Supercomputers use Linux as their host OS.
- You can also find Linux distros in:
  - Routers, firewalls, switches
  - Smartphones (see Android)
  - Gaming consoles (Sony PlayStation, Valve SteamBox)



Graph showing the market share of Linux in the SuperComputer area

Know the Difference Between Linux and Windows

LINUX	WINDOWS
Free & Open source	Paid & not an open source
Case Sensitive	Case In-sensitive
Monolithic Kernel	Micro Kernel
Hierarchical file system	Flat file system
Command line / Terminal	Has Commandline but is not as functional as linux command line
Licensing Freedom	Licensing restriction
Full access on OS	No full access on OS
More Secure than windows	Less secure than Linux

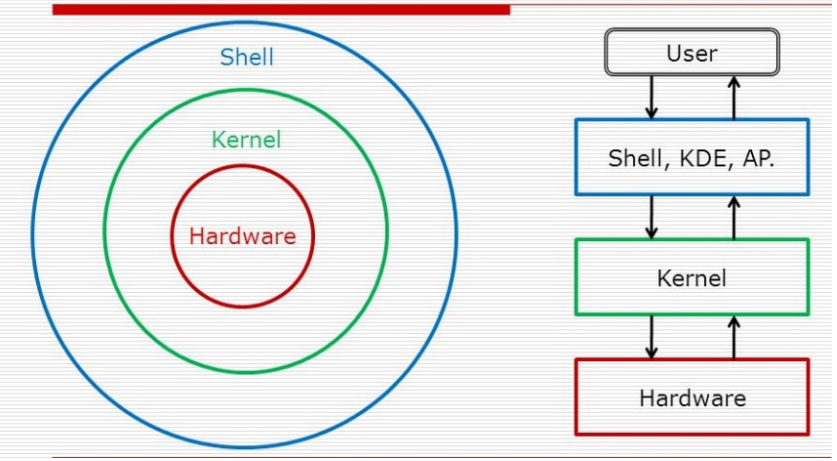
Simplified architecture of Linux (I)

Kernel:

- The kernel is the heart of the operating system.
- It interacts with hardware and most of the tasks like memory management, task scheduling and file management.

Shell:

- The shell is the utility that processes your requests.
- When you type in a command at your terminal, the shell interprets the command and calls the program that you want.
- The shell uses standard syntax for all commands.
- C Shell, Bourne Shell and Korn Shell are most famous shells which are available with most of the Unix variants.



Simplified architecture of Linux (II)

Commands and Utilities:

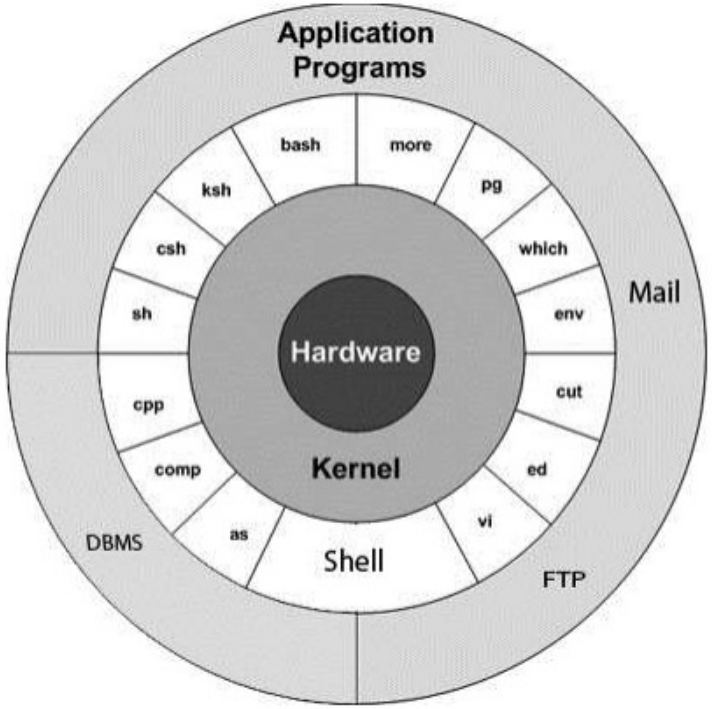
- There are various commands and utilities which you would use in your day to day activities.
- **cp**, **mv**, **cat** and **grep** are few examples of commands and utilities.
- There are over 250 standard commands plus numerous others provided through 3rd party software.
- All the commands come along with various optional options.

Files and Directories:

- All data in Linux is organized into files. All files are organized into directories.
- These directories are organized into a tree-like structure called the filesystem.

Simplified architecture of Linux (III)

The diagram:

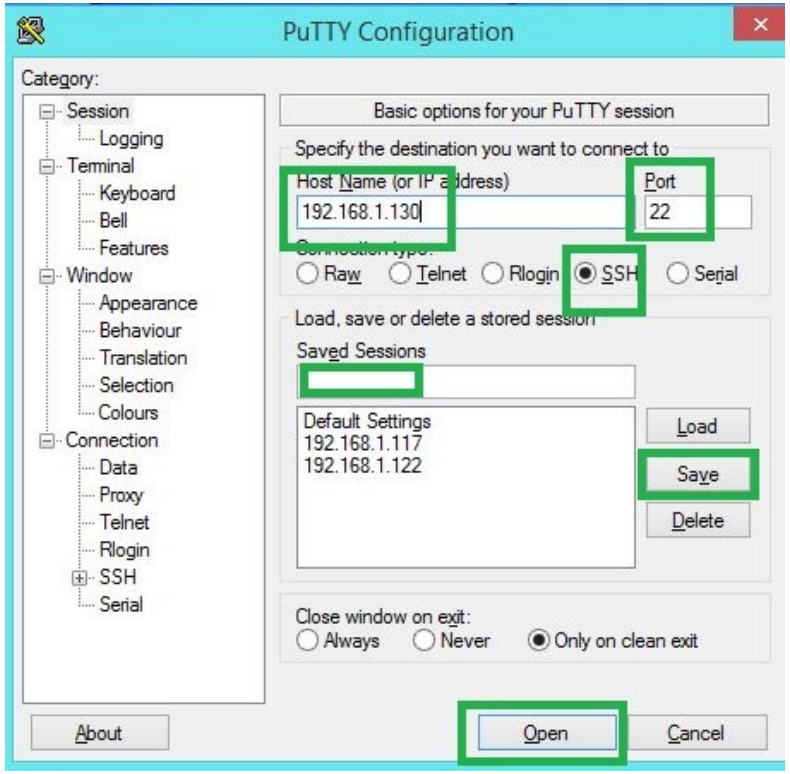


Remote access to a linux server (I)

- Usually is done via SSH
  - ssh host - Connect to host as your local username.
  - ssh user@host - Connect to host as user
  - ssh -p port user@host - Connect to host using port
- The SSH server can be installed like this:
  - sudo apt-get install openssh-server // in Ubuntu
  - yum install openssh-server // In RedHat, CentOS
- Start the SSH server:
  - sudo service ssh restart // in Ubuntu
  - service sshd start //in Redhat, CentOS
- Download a terminal emulator client:
  - putty or Ericom Interconnect

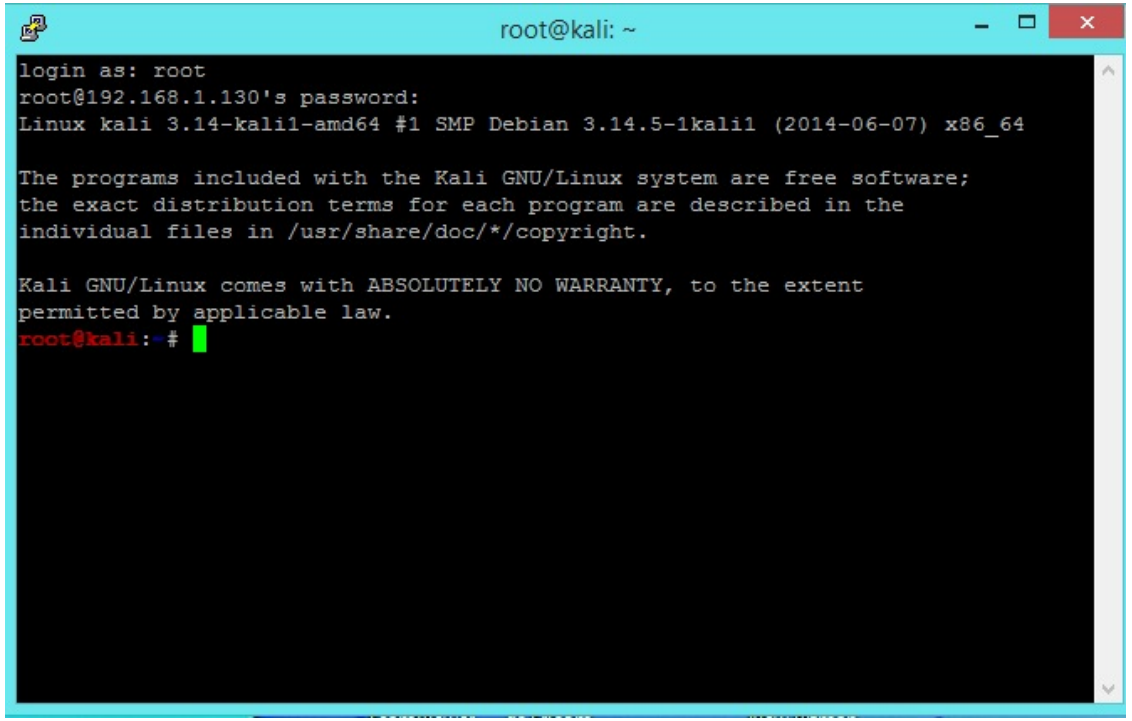
Remote access to a linux server (II)

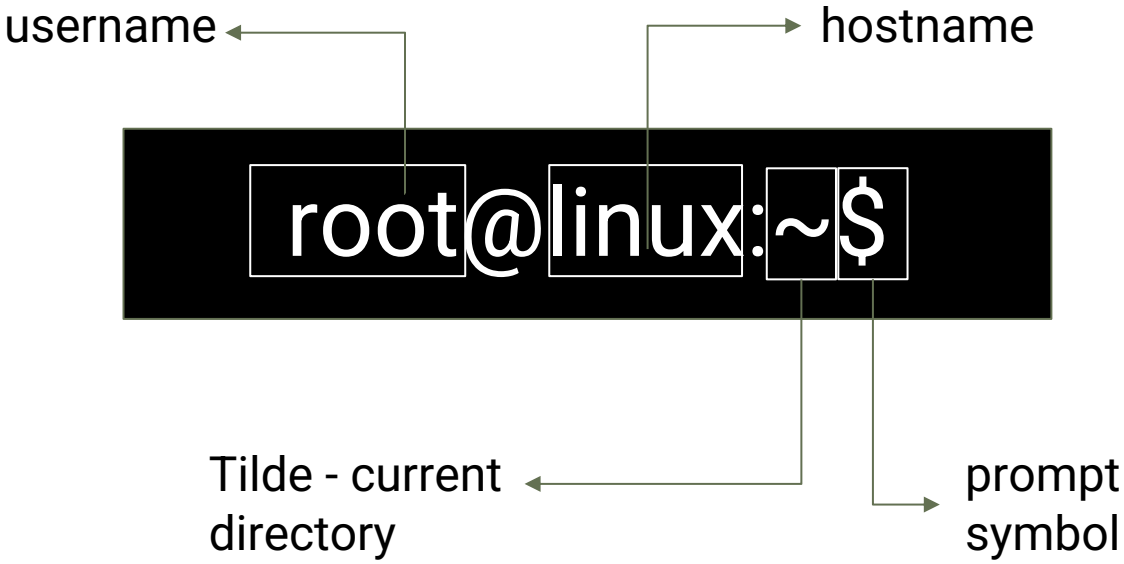
Run putty, enter the hostname/IP, the port (default is 22) and hit “Open”.



Remote access to a linux server (III)

Enter the user/password and you are connected to the Linux BASH environment





BASH – the Linux shell

- BASH is a programming/scripting language
- BASH shell is the Linux equivalent of the Windows **cmd**
- BASH is a command processor that typically runs in a text window, where the user types commands that cause actions
- BASH runs scripts (python, perl, etc)
- It has been ported to Windows (via Cygwin)

BASH – the golden rule

- When you do not know what a command does:
  - **man** – stands for manual
    - man ls
    - man cd
    - man grep
    - etc,

System information (II)

So you are logged into this black Linux shell, but you have no info about the type of Linux distro or the architecture…

- **uname** – prints the name, version and other details about the current machine and the operating system running on it.
- **uname -a -** Display Linux system information
- **uname -r -** Display kernel release information
- **hostname** - Show system host name
- **hostname -I** - Display all local IP addresses of the host.
- **Reboot** - Show system reboot history last
- **Date** - Show the current date and time
- **cal** - Show this month's calendar

System Information (II)

- **whoami** – shows the user you are currently logged in with
- **users** – displays (all) the users currently logged in
- **cat /etc/redhat-release** - Show which version of Red Hat installed
- **lsb_release -a** – prints ubuntu Distribution information.
- uptime - Show how long the system has been running + load
- w Display who is online
- **Whoami** - Who you are logged in as

Hardware information (II)

Using tools like lscpu as lscpu is an easy way to get CPU information

- lscpu - CPU architecture information from sysfs, /proc/cpuinfo and any applicable architecture-specific libraries
- **lshw | grep** cpu - **to display a complete picture of hardware configuration**
- **lspci -tv** - Display PCI devices lspci -tv
- **lsusb -tv** - Display USB devices
- **cat /proc/cpuinfo** - Display CPU information
- **cat /proc/meminfo** - Display memory information
- **grep -c processor /proc/cpuinfo -** count processor (including cores)

Hardware information (II)

- **dmesg** - Display messages in kernel ring buffer
- **free -h** - Display free and used memory (-h for human readable, -m for MB, -g for GB.)
- **dmidecode** - Display DMI/SMBIOS (hardware info) from the BIOS
- **sda hdparm -i /dev/sda** - Show info about disk# Perform a read speed test on disk
- **sda hdparm -tT /dev/sda** - Test for unreadable blocks on disk sda badblocks -s /dev/sda

Copying, renaming and deleting files

Make a **copy** of a file using the **cp** command.

- cp source_file destination_file

- Renaming a file with the **mv** command:
  - mv old_file new_file

- Delete one or multiple files with **rm**:
  - rm filename1
  - rm filename1 filename2 filename3 //multiple files
  - rm –r –f /home/cristian/* // deletes all files in /home/Cristian without confirmation
  - rm *.txt ./ //deletes all .txt files in the current directory

Change ,Create & Remove directory

- **cd** - The cd command is used to change the current directory (i.e., the directory in which the user is currently working) in Linux.
  - cd /home/cristian
  - cd ~ // “~” stands for the user’s home directory

- **mkdir** - The mkdir command is used to create the new directory
  - mkdir mydirectory
  - mkdir -p mydir2/mysubdir2/threedirsdeep

- **rmdir** - The rmdir command is used to delete the directory
  - rmdir mydirectory. -- Can be used when the directory is empty
  - mkdir -p mydirectory. -- can be used to recursively remove

File Management (I)

ls - list directory contents

- ls –lh /home/
- ls ./

ls -a	list all files including hidden file starting with '.'
ls --color	colored list [=always/never/auto]
ls -d	list directories - with ' */'
ls -i	list file's inode index number
ls -l	list with long format - show permissions
ls -la	list long format including hidden files
ls -lh	list long format with readable file size
ls -ls	list with long format with file size
ls -r	list in reverse order
ls -R	list recursively directory tree
ls -s	list file size
ls -S	sort by file size
ls -t	sort by time & date
ls -X	sort by extension name

File Management (II)

cat - (short for concatenate) command is one of the most frequently used commands on Linux

It can be used for:

- Display text file on screen
- Read text file
- Create a new text file
- Modifying file

File Management (III)

- **Read text file**

  cat file_name

  cat /path/to/file

```
[root@fcsteaua ~]# cat /root/blockip.sh
#!/bin/bash
BLOCKDB="/root/ip.blocked"
IPS=$(grep -Ev "^#" $BLOCKDB)
for i in $IPS
do
iptables -A INPUT -s $i -j DROP
iptables -A OUTPUT -d $i -j DROP
done
[root@fcsteaua ~]#
```

- **Create a new text file**
  - cat > newfile.txt // can be done with the **touch** command

```
[root@fcsteaua ~]# cat > foo.txt
I'm inserting some text here
pressing enter to go to the next line
and pressing Ctrl+D to save and exit
[root@fcsteaua ~]#
```

File Management (IV)

- Create symbolic link to linkname
  ln -s /path/to/file linkname
- Create an empty file or update the access and modification times of file.
  touch file
- View the contents of file cat file # Browse through a text file
  less file
- Display the first 10 lines of file
  head file
- Display the last 10 lines of file
  tail file
- Display the last 10 lines of file and "follow" the file as it grows.
  tail -f file

File Management (V)

- **Modifying file:**
  - To append (add data to existing) data to a file called foo.txt, enter:

```
[root@fcsteaua ~]# cat >> foo.txt
I need to add some new text to this file
the existing text in the file will
not be overwritten
[root@fcsteaua ~]#
```

- **Extra:**
  - List the foo.txt file and display line numbers
  - Very useful when you encounter script errors

```
[root@fcsteaua ~]# cat -n foo.txt
     1  I'm inserting some text here
     2  pressing enter to go to the next line
     3  and pressing Ctrl+D to save and exit
     4  I need to add some new text to this file
     5  the existing text in the file will
     6  not be overwritten
[root@fcsteaua ~]#
```

- **FILE:**
  - This command determines the file type
  - File <filename>. , file –s /dev/sda

Sort,Uniq,Comm,cmp & Diff

- sort – this filter will default to an alphabetical sort
  - Sort <filename>
  - Sort –k1 , sort –k2 ..

- Uniq – to remove duplicated from a sorted list
  - Sort <filename> | uniq
  - Sort <filename> | uniq –c

- Comm - Comparing streams (or files) can be done with the **comm**. By default **comm** will output three columns**.**
  - Comm file1 file2
  - Comm -12 file1 file2
  - Comm -23 file1 file2

- Cmp – This is used to compare two files byte by byte. If a difference is found, it reports the byte and line number where the first difference is found. If no differences are found, by default, **cmp** returns no output
  - Cmp file1 file2

- Diff - diff stands for **difference**. This command is used to display the differences in the files by comparing the files line by line
  - Diff file1 file2
    - Special symbols are:
    - a : add c : change d : delete

Input and Output Redirection

- **Redirecting the input:**
  command < filename
  Any input will read from that file (must be file).
- **Redirecting the output.**
  - The simple > rewrites the output file, while the double one >> appends to the file (must be file).
    command > filename
    command >> filename
- **Combine input and ouput redirections**
- **Example:**
  - wc < my_text_file.txt > output_file.txt

Filter output with **grep**

grep - searches the named input FILEs for the lines that match the specified pattern

- grep is the equivalent of **findstr.exe** in Windows

- Example:
  - I want to list the /var/log/messages file for the “error” patern
    - **grep error /var/log/messages**

```
[root@fcsteaua ~]# grep error /var/log/messages
Jun 19 09:43:05 fcsteaua proftpd[30145]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01
:4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:05 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01
:4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:08 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01
```

    - Or with pretty colors
    - **grep -i error /var/log/messages**

```
[root@fcsteaua ~]# grep --color error /var/log/messages
Jun 19 09:43:05 fcsteaua proftpd[30145]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:05 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:08 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:08 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
```

Control the output with **more** & **less**

- **more** - is a filter for paging through text one screenful at a time
- **less** - is a program similar to more (1), but which allows backward movement in the file as well as forward movement.
  - The syntax:
    - **more** /my/log/file
    - **less** /my/log/file

```
Jun 14 03:16:44 fcsteaua rsyslogd: [origin software="rsyslogd" swVersion="5.8.10"
 x-pid="24531" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
Jun 14 03:37:37 fcsteaua xinetd[1505]: START: smtp pid=8376 from=::ffff:123.28.30
.213
Jun 14 03:37:41 fcsteaua xinetd[1505]: EXIT: smtp status=1 pid=8376 duration=4(se
c)
Jun 14 03:42:52 fcsteaua xinetd[1505]: START: smtp pid=9001 from=::ffff:123.28.30
.213
Jun 14 03:42:56 fcsteaua xinetd[1505]: EXIT: smtp status=1 pid=9001 duration=4(se
c)
Jun 14 05:36:25 fcsteaua xinetd[1505]: START: smtp pid=18207 from=::ffff:171.232.
134.128
Jun 14 05:36:30 fcsteaua xinetd[1505]: EXIT: smtp status=1 pid=18207 duration=5(s
ec)
Jun 14 06:19:16 fcsteaua xinetd[1505]: START: ftp pid=21848 from=::ffff:135.196.3
7.137
Jun 14 06:19:16 fcsteaua proftpd[21848]: processing configuration directory '/etc
/proftpd.d'
Jun 14 06:19:16 fcsteaua proftpd[21848]: 136.243.1.30 (135.196.37.137[135.196.37.
137]) - FTP session opened.
Jun 14 06:19:17 fcsteaua proftpd[21848]: 136.243.1.30 (135.196.37.137[135.196.37.
137]) - FTP session closed.
Jun 14 06:19:17 fcsteaua xinetd[1505]: EXIT: ftp status=0 pid=21848 duration=1(se
c)
/var/log/messages
```

Search(I)

- **locate -** locate a file
  - locate mykey.pem

combine the **locate** command with **grep** using a pipe like this:

  - **locate** pem **| grep** mykey

- **find /** -name mykey.pem -print
  - **find** utility can do much more than find files, but a full description of its capabilities is beyond the scope of this example

Search (II)

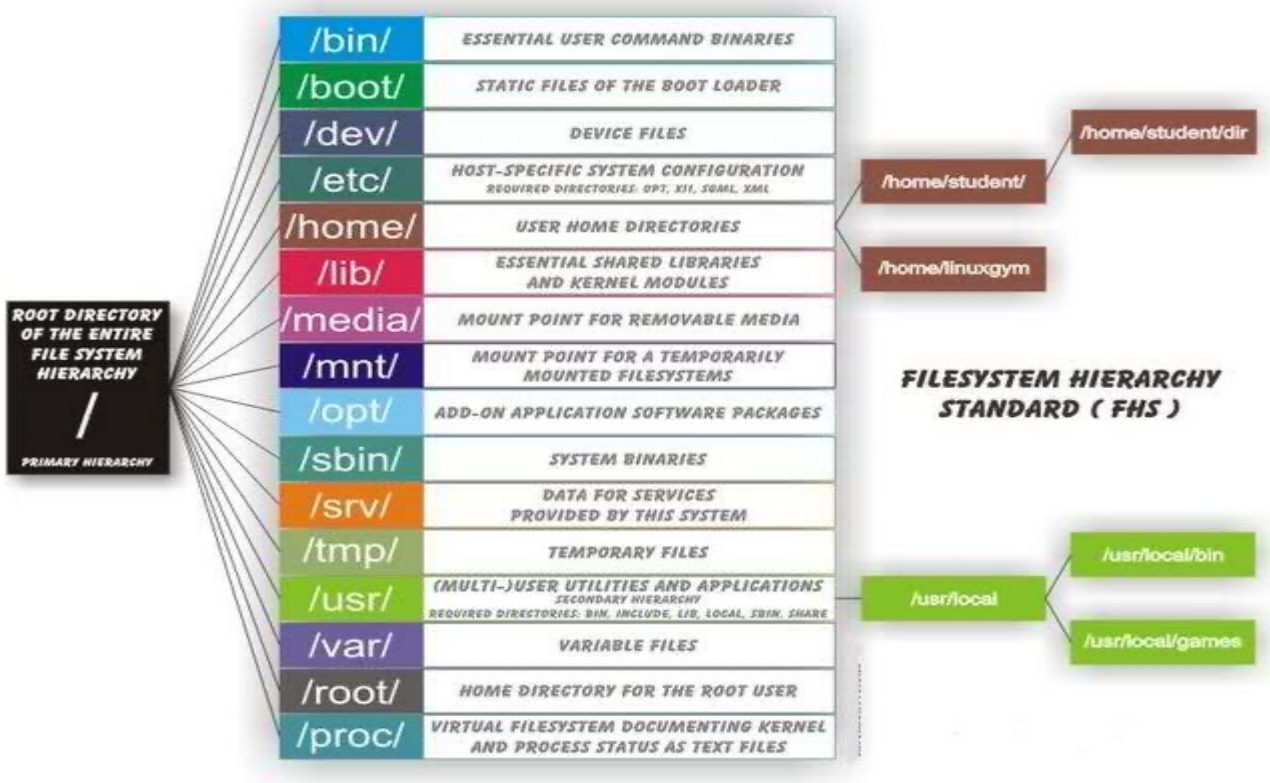
- Search for pattern in file grep pattern file
  grep pattern file
- Search recursively for pattern in directory
  grep -r pattern directory
- Find files in /home/john that start with "prefix".
  find /home/john -name 'prefix*’
- Find files larger than 100MB in /home
  find /home -size +100M

File Transfers

- Secure copy file.txt to the /tmp folder on server
  scp file.txt server:/tmp
- Copy *.html files from server to the local /tmp folder.
  scp server:/var/www/*.html /tmp
- Copy all files and directories recursively from server to the current system's /tmp folder.
  scp -r server:/var/www /tmp
- Synchronize /home to /backups/home
  rsync -a /home /backups/
- Synchronize files/directories between the local and remote system with compression enabled
  rsync -avz /home server:/backups/

Linux directory hierarchy

- In Windows we call them folders, in Linux the term used is **directory/directories.**



Linux directory hierarchy (II)

- The equivalent of the “C:\” partition in Windows is referred in Linux as “**/**” – also called “root directory”, or “slash”.
- The Linux filesystem has the root directory at the top of the directory tree.
- The following list of directories are subdirectories of the root directory. This directory is denoted by the / (pronounced "slash") symbol.
  - **/bin:**
    Contains executable programs such as ls (“dir” in Windows) and cp (“copy” in Windows). These programs are designed to make the system usable.
  - **/etc**
    Contains configuration files which are local to the machine. Programs store configuration files in this directory and these files are referenced when programs are run.
  - **/home**
    Contains user account directories. Each user created by the system administrator will have a subdirectory under /home with the name of the account. This is the default behaviour of Linux systems. E.g. User account for Anna is created, her home directory will be located in /home/anna.

Linux directory hierarchy (III)

- **/mnt**
  Used for mounting temporary filesystems. When mounting a CD-ROM for instance, the standard mount point location is /mnt/cdrom.
- **/opt**
  Used for storing random data that has no other logical destination.
- **/proc**
  Provides information about running processes and the kernel. A directory is provided for each running process. Useful system information such as the amount of Random Access Memory (RAM) available on the system as well as Central Processing Unit (CPU) speed in Megahertz (MHz) can be found within the /proc directory.
- **/root**
  This is the home directory for the super user (root). This directory is not viewable from user accounts. The /root directory usually contains system administration files.
- **/sbin**
  Similar to /bin, this directory contains executable programs needed to boot the system, however the programs within /sbin are executed by the root user.
- **/tmp**
  This directory is used for temporary storage space. Files within this directory are often cleaned out either at boot time or by a regular job process.

Linux directory hierarchy (IV)

- **/usr**
  Used to store applications. When installing an application on a Debian GNU/Linux machine, the typical path to install would be /usr/local. You will notice the directory structure within /usr appears similar to the root directory structure.

- **/var**
  This directory contains files of variable file storage. Files in /var are dynamic and are constantly being written to or changed. This the directory where websites are usually stored in.

Performance Monitoring And Statistics (I)

Hard drive usage:

- **df** - displays the amount of disk space available on the file system

  df -h

  watch df –h - showing periodic updates

du - estimates and displays the disk space used by files and directories

```
[root@fcsteaua ~]# du -h /sbin/
12M     /sbin/
[root@fcsteaua ~]#
```

- Display free and used memory (-h for human readable, -m for MB, -g for GB.)

  free -h

Performance Monitoring And Statistics (II)

CPU

To get processors related statistics you can use mpstat command but with some options it will provide better visibility:

$ mpstat 2 10

Memory

We all know command **free** to show amount of (remaining) RAM but to see all statistic including I/O operations:

$ **vmstat** 2 10

Disk

To get general information about your disk operations in real time you can utilise iostat.

$ iostat -kx 2

Performance Monitoring And Statistics (III)

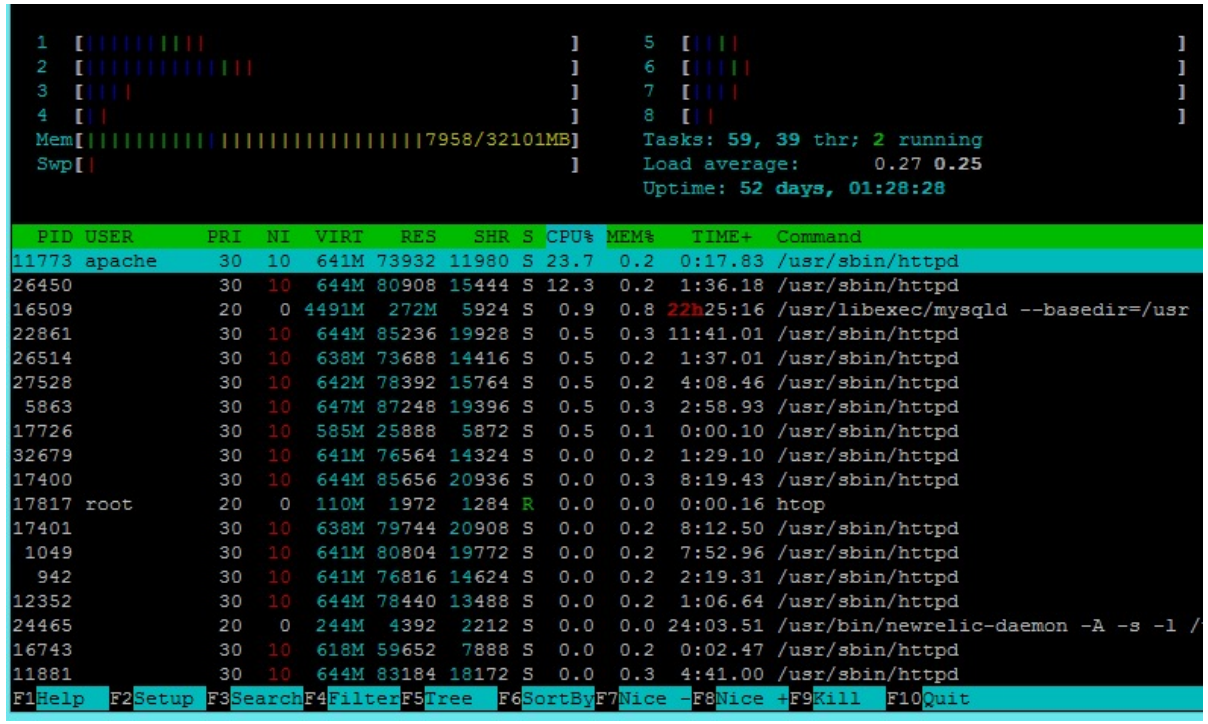
lsof - a command meaning "list open files", which is used in many Unix-like systems to report a list of all open files and the processes that opened them.

```
[root@fcsteaua log]# lsof -i
COMMAND     PID         USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
qmail-smt   583       qmaild    0u  IPv4 100264079      0t0  TCP fcsteaua.ro:smtp->118.179.212.50:53799 (ESTA
ISHED)
qmail-smt   583       qmaild    1u  IPv4 100264079      0t0  TCP fcsteaua.ro:smtp->118.179.212.50:53799 (ESTA
ISHED)
qmail-smt   583       qmaild    2u  IPv4 100264079      0t0  TCP fcsteaua.ro:smtp->118.179.212.50:53799 (ESTA
ISHED)
xinetd     1505         root    5u  IPv6  10088076      0t0  TCP *:urd (LISTEN)
xinetd     1505         root    6u  IPv6      9868      0t0  TCP *:poppassd (LISTEN)
xinetd     1505         root    8u  IPv6      9869      0t0  TCP *:smtp (LISTEN)
xinetd     1505         root    9u  IPv6   2643891      0t0  TCP *:ftp (LISTEN)
xinetd     1505         root   10u  IPv6      9871      0t0  TCP *:submission (LISTEN)
sw-cp-ser  1975         root    7u  IPv4     11120      0t0  TCP *:6308 (LISTEN)
sw-cp-ser  1975         root    8u  IPv4     11121      0t0  TCP *:pcsync-https (LISTEN)
sw-cp-ser  1975         root    9u  IPv4     11122      0t0  TCP *:cddbp-alt (LISTEN)
sw-cp-ser  1975         root   10u  IPv6     11123      0t0  TCP *:pcsync-https (LISTEN)
sw-cp-ser  1975         root   11u  IPv6     11124      0t0  TCP *:cddbp-alt (LISTEN)
sw-cp-ser  1976 sw-cp-server    7u  IPv4     11120      0t0  TCP *:6308 (LISTEN)
sw-cp-ser  1976 sw-cp-server    8u  IPv4     11121      0t0  TCP *:pcsync-https (LISTEN)
sw-cp-ser  1976 sw-cp-server    9u  IPv4     11122      0t0  TCP *:cddbp-alt (LISTEN)
sw-cp-ser  1976 sw-cp-server   10u  IPv6     11123      0t0  TCP *:pcsync-https (LISTEN)
sw-cp-ser  1976 sw-cp-server   11u  IPv6     11124      0t0  TCP *:cddbp-alt (LISTEN)
httpd      5800       apache    4u  IPv6  60496294      0t0  TCP *:http (LISTEN)
httpd      5800       apache    6u  IPv6  60496298      0t0  TCP *:https (LISTEN)
httpd     10923         root    4u  IPv6  60496294      0t0  TCP *:http (LISTEN)
```

Process Management (I)

- Processor, memory, general server load
  - **top -** provides a dynamic real-time view of a running system. It can display system summary information, as well as a list of processes or threads currently being managed by the kernel

- Processor, memory, general server load
  - **htop –** similar to top, but with more details and fancier colors



Process Management (II)

- Display your currently running processes
  ps
- Display all the currently running processes on the system.
  ps -ef
- Display process information for processname
  ps -ef | grep processname

Process Management (III)

- Kill process with process ID of pid

  kill pid
- Kill all processes named processname

  killall processname
- Start program in the background

  program &
- Display stopped or background jobs

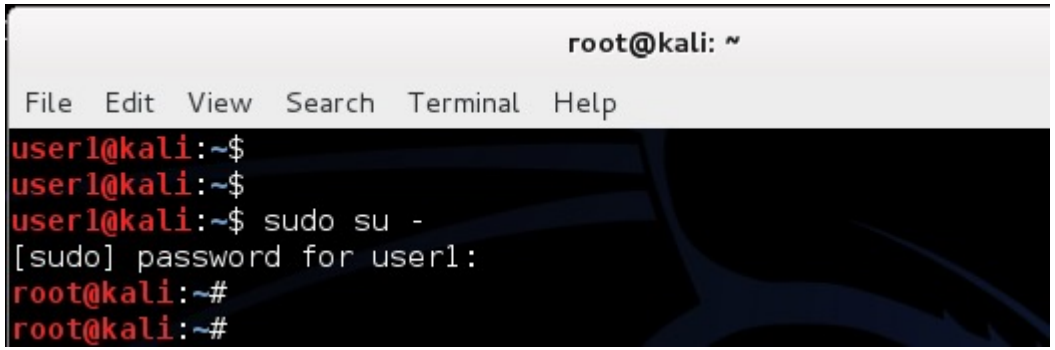
  bg
- Brings the most recent background job to foreground

  fg
- Brings job n to the foreground

  fg n

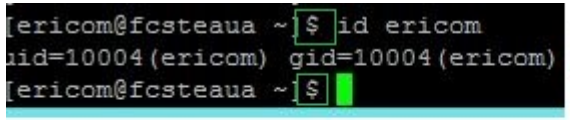
Users and groups

- Similar to Windows:
  - Linux has limited access users and, by default, one administrator (called “**root**”)
  - **root** is the user name or account that by default has access to all commands and files on Linux.
  - It is also referred to as the root account, root user and the superuser.
  - You can grant root like access to limited users using **sudo** (see “Run as Administrator in Windows”)



Users and groups (II)

- With **sudo**, as a limited permissions user, you can be granted, temporarily, administrator/root access to execute commands usually restricted to only the root user.
- **sudo** is used in Linux Debian derivatives distros (Ubuntu, SteamOS from Valve, Kali Linux, etc) – but not limited to only Debian
- **sudo** cand be installed on any Linux system
- Not every user can use sudo. That user must be present in the /etc/sudoers file
- In the BASH environment/the linux shell, the root user can be recognized by
  - the pound sign (**#**). Limited users can be recognized by the “$” sign after their name.
  - When not sure about the user you are currently logged in, issue the **whoami** command



I am not root.
No # sign after my name

Users and groups (III)

- All users have:
  - user IDs (**uid**), group IDs (**gid**).
  - The **uid** and **gid** are always decimal numbers and start from 1000 or 10000
  - The root superuser usually has **uid** and **gid** 0 (zero)
  - A specific user can be member of multiple groups.
- The **id** command show all the information you need to know about a user
- Try issuing the **id root** command and see what happens

```
[root@fcsteaua ~]# id cristi
uid=10003(cristi) gid=10003(cristi) groups=10003(cristi)
[root@fcsteaua ~]#
```

Users and groups (IV)

- How do I **add a new user** via the linux shell?
  - useradd Cristian –p test123

  The command above created a new user called ericom with the password test123

- How do I **assign a user to another group**?
  - usermod –G root Cristian
  - sudo usermod –aG guest
- I added the **user Cristian** to the **root group**.

Users and groups (V)

- Create a new group:

```
[root@fcsteaua ~]# groupadd connect-group
[root@fcsteaua ~]# cat /etc/group | grep connect
connect-group:x:10005:
[root@fcsteaua ~]#
```

- Delete a group:

```
[root@fcsteaua ~]# groupdel connect-group
[root@fcsteaua ~]# cat /etc/group | grep connect
[root@fcsteaua ~]#
```

Users and groups (VI)

- Change the password of a user with the **passwd** command:

```
[root@fcsteaua ~]# passwd ericom
Changing password for user ericom.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@fcsteaua ~]#
```

  - Login as root if you are changing a password for an account different than yours
    - If you are logged in with a limited user account, use the **su** command or **sudo su** to login as root

```
bash-4.1$ id cristi
uid=10003(cristi) gid=10003(cristi) groups=10003(cristi)
bash-4.1$ su root
Password:
[root@fcsteaua ~]# id
uid=0(root) gid=10004(ericom) groups=10004(ericom),0(root),1
s),4(adm),6(disk),10(wheel)
[root@fcsteaua ~]#
```

<< Limited user

<< become root

<< Confirming that I am the root superuser

Permissions system in Linux (I)

Each file and directory has three user based permission groups:

- **owner** - The Owner permissions apply only the owner of the file or directory, they will not impact the actions of other users.
- **group** - The Group permissions apply only to the group that has been assigned to the file or directory, they will not effect the actions of other users.
- **all users** - The All Users permissions apply to all other users on the system, this is the permission group that you want to watch the most.

Permissions system in Linux (II)

Permission Types

Each file or directory has three basic permission types:

- The **read permission** grants the ability to read a file. When set for a directory, this permission grants the ability to read the names of files in the directory, but not to find out any further information about them such as contents, file type, size, ownership, permissions.
- The **write permission** grants the ability to modify a file. When set for a directory, this permission grants the ability to modify entries in the directory. This includes creating files, deleting files, and renaming files.
- The **execute permission** grants the ability to execute a file. This permission must be set for executable programs, including shell scripts, in order to allow the operating system to run them. When set for a directory, this permission grants the ability to access file contents and meta-information if its name is known, but not list files inside the directory, unless read is set also

Permissions system in Linux (III)

- View the permissions:
  - **ls** is the utility you need
  - Is the equivalent of **dir** in Windows
  - Standard usage is ls –lh (list, show permissions and display them in human readable format)
  - Any file or folder that starts with a dot character (for example, /home/user**/.config**), commonly called a dot file or dotfile, is hidden.

```
[root@fcsteaua ericom]# ls -lh
total 0
-rw-r--r-- 1 root ericom 0 Jun 12 16:02 visible-file.txt
[root@fcsteaua ericom]#
[root@fcsteaua ericom]# ls -alh
total 20K
drwx------  2 ericom ericom 4.0K Jun 12 16:02 .
drwxr-xr-x. 7 root   root   4.0K Jun 12 14:47 ..
-rw-r--r--  1 ericom ericom   18 Oct 16  2014 .bash_logout
-rw-r--r--  1 ericom ericom  176 Oct 16  2014 .bash_profile
-rw-r--r--  1 ericom ericom  124 Oct 16  2014 .bashrc
-rw-r--r--  1 root   ericom    0 Jun 12 16:02 .invisible-file.txt
-rw-r--r--  1 root   ericom    0 Jun 12 16:02 visible-file.txt
[root@fcsteaua ericom]#
```

<< List only non-hidden files

<< List non-hidden AND hidden files

Permissions system in Linux (IV)

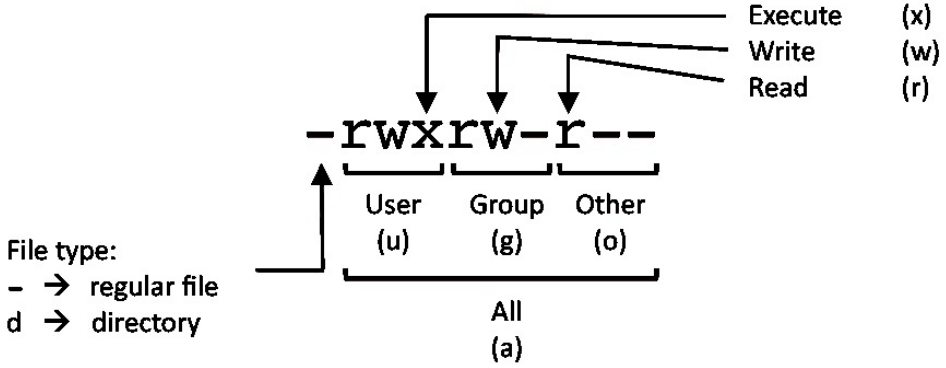
- Reading the file and directory permissions

  -rw-r--r-- 1 root ericom 0 Jun 12 16:02 file.txt

- The first character (-) indicates the file type and is not related to permissions. The remaining nine characters are in three sets, each representing a class of permissions as three characters:
  - The **first** set represents the **user class.**
  - The **second** set represents the **group class.**
  - The **third** set represents the **others class**.

Each of the three characters represent the read, write, and execute permissions:

- r if reading is permitted, - if it is not.
- w if writing is permitted, - if it is not.
- x if execution is permitted, - if it is not.



- In our example, **-rw-r--r-- root ericom** means:

rw-	r--	r--
The owner (root) can read and write the file	The users in the ericom group can read the file	Everyone else can read the file

Permissions system in Linux (V)

Another example:

-rwxr-x--- 1 root ericom 144K Jun 12 11:02 script.sh

rwx	**r-x**	**---**
Owner (root in this case) can read, write and execute the file	The users in the ericom group can read and execute the file	Everyone else cannot read, write or execute the files.

r	4
w	2
x	1

-(rwx)	(rwx)	(r--)
-(421)	(421)	(4--)
7	7	4

Read,write and execute permissions for user and group. Read permissions for others

Permissions system in Linux (VI)

The alternative to the symbolic (rwx) permission system:

Meet the octal notation:

Symbolic Notation	Octal Notation	Number	English
----------	0000	0	no permissions
---x--x--x	0111	1	execute
--w--w--w-	0222	2	write
--wx-wx-wx	0333	3	write & execute
-r--r--r--	0444	4	read
-r-xr-xr-x	0555	5	read & execute
-rw-rw-rw-	0666	6	read & write
-rwxrwxrwx	0777	7	read, write, & execute

Permissions system in Linux (VII)

Modify the permissions with **chmod**

When you:

- grant permission you use the plus sign “+”
- take permission away you will use the minus sign “-”

Example 1:
Grant permission for read, write and execute to the file owner
chmod u+rwx file.txt //in octal: chmod 700 file.txt

Example 2:
Take away all privileges from user eircom for file.txt
chmod u-rwx file.txt

Permissions system in Linux (VIII)

Example 3:

Grant permission for read, write and execute for user, group and everyone else

chmod ugo+rwx file.txt // in octal: chmod 777 file.txt

Example 2:

Take away all privileges from user, group and everyone else

chmod ugo-rwx file.txt // in octal: chmod 000 file.txt

Example 3:

Grant recursive permission in a specific directory

chmod -R ugo+rwx /path/to/my/directory // in octal: chmod –R 777 /path/to/my/directory

Environment Variables

The environment variables are global variables that are defined within the shell and that can be used for many purposes. Their names are in general in all uppercase.

- Display the value of an environment variable:
  echo $PATH

- $PATH - Lookup list for binaries to be executed.

- $HOME - user's home directory

env command allows you to set or print the environment variables. During troubleshooting, you may find it useful for checking if the wrong environment variable prevents your application from starting

How do I install software using a package manager?

In Debian & Ubuntu like systems:

apt-get install apache2

// installs the Apache httpd server

```
root@kali:~# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libafpclient0 libcrypt-passwdmd5-perl libgadu3 libmozjs24d xulrunner-24.0
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  apache2
```

In Redhat and CentOS like systems:

yum install httpd

//installs Apache httpd server. See the difference in names!

RPMs and DEB files

RPM Package Manager (**RPM**) (originally Red Hat **Package** Manager) is a **package** management system. The name **RPM** variously refers to the .**rpm file** format, **files** in this format, software packaged in such **files**, and the **package** manager itself.

deb is the extension of the Debian software **package** format and the most often used name for such binary **packages.**

How do I install software without a package manager (I)?

In Debian & Ubuntu like systems:

wget http://www.eu.apache.org/dist//directory/apacheds/dist/2.0.0-M20/apacheds-2.0.0-M20-amd64.deb

//download the file DEB file

chmod +x apacheds-2.0.0-M20-amd64.deb // make the file executable

dpkg -i apacheds-2.0.0-M20-amd64.deb // install the Apache DEB package

/etc/init.d/apache2 start ///start Apache

How do I install software without a package manager (II)?

In Redhat and CentOS like systems:

wget ftp://rpmfind.net/linux/centos/5.11/os/i386/CentOS/httpd-2.2.3-91.el5.centos.i386.rpm

//download the RPM file

chmod+x httpd-2.2.3-91.el5.centos.i386.rpm // make the file executable

rpm -i httpd-2.2.3-91.el5.centos.i386.rpm // install the httpd RPM file

service httpd start // start the Apache server

How do I install software by compiling from the source (I)?

- Software can be installed from the code source without being a developer
- You need root access or you can use **sudo**
- You will need a C compiler (called GCC in Linux)
- Access to a BASH console is mandatory

How do I install software by compiling from the source (II)?

Example. Install **pidgin** from source code in Ubuntu.

- **sudo apt-get install build-essential** // this will install the compiler and other required libraries

Now you'll need your desired application's source code. These packages are usually in compressed files with the .tar.gz or .tar.bz2 file extensions.

- **wget** http://downloads.sourceforge.net/project/pidgin/Pidgin/2.10.11/pidgin-2.10.11.tar.bz2
- **tar** -xjvf pidgin-2.10.11.tar.bz2 // extract the content of the archive
- **cd** pidgin-2.10.11 // navigate to the new created directory
- **./configure** // **configure the new install**
- **make** // compile the program
- **make install** // install the software on your system

Known Linux server applications

- HTTP server:
  - Apache (httpd), nginx
- SQL:
  - Mysql (mysqld), SQLite, postgresql
- FTP servers:
  - Proftpd, Pure-FTPd, vsFTPd, Filezilla
- DNS servers (Bind),
- Firewall (iptables, ipchains),
- SMTP servers (postfix, qmail, sendmail),
- POP3 / IMAP servers (Dovecot, Courier)
- Remote access server (OpenSSH)

Known Linux applications (I)

- **Text editors**
  - vi

Vi is a powerful text editor included with most Linux systems, even embedded ones. Sometimes you'll have to edit a text file on a system that doesn't include a friendlier text editor, so knowing Vi is essential.

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export LS_OPTIONS="--human --color=always"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -Al'
```

Known Linux applications (II)

- **Text editors**
  - nano

nano is a small and friendly text editor. Besides basic text editing, nano offers many extra features like an interactive search and replace, go to line and column number.

```
  GNU nano 2.0.9              File: .bashrc

# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export LS_OPTIONS="--human --color=always"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -Al'
#export wgets='H="--header"; wget $H="Accept-Language: en-us,en;q=0.5" $H="Acce$

                              [ Read 18 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Known Linux applications (III)

- **Text editors**
  - emacs

Emacs is one of the oldest and most versatile text editors available for Linux and UNIX-based systems. It's been around for a long time and is well known for its powerful and rich editing features.

```
File Edit Options Buffers Tools Sh-Script Help
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

export LS_OPTIONS="--human --color=always"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -Al'
#export wgets='H="--header"; wget $H="Accept-Language: en-us,en;q=0.5" $H="Acce\
pt: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" $H="Connec\
tion: keep-alive" -U "Mozilla/5.0 (Windows NT 5.1; rv:10.0.2) Gecko/20100101 Fi\
refox/10.0.2" --referer=/ '
-UU-:----F1  .bashrc        All L1     (Shell-script[bash])----------------------
Indentation setup for shell type bash
```

How to run Linux scripts

- You have the blockip.sh script that is located /home/eircom
- First check if the script can be executed by the user you are currently logged in with:
  - ls -lh /home/ericom/blockip.sh
- If you cannot execute it, do a:
  - **chmod u+rx /home/cristi/blockip.sh** // or **chmod 500 /home/cristi/blockip.sh**
- Run the script:
  - **/home/ericom/blockip.sh** // or if you are already in the /home/eircom, run it with **./blockip.sh**
  - If your connection drops your scipt might crash
- Make the script run after you exit the shell or the connection is interrupted:
  - **nohup /home/cristi/blockip.sh &** // hit enter twice

Creating and Extracting archives (I)

- Most seen file extensions are **.tar.gz** and **.tar.bz2** which is a tar archive further compressed using **gzip** or **bzip** algorithms respectively.
- Create archives
  - tar -cvf mynewarchive.tar /var/www
  - (will create mynewarchive.tar with the content of /var/www)
  - tar -czvf file.tar.gz directory
  - tar -czvf filename.tar.gz /path/to/dir1
  - tar -czvf filename.tar.gz /path/to/dir1 dir2 file1 file2
  - # Create a tar.gz file from all pdf (".pdf") files
    tar -czvf archive.tgz *.pdf
- Extract a tar.gz archive:
  - tar -xvzf tarfile.tar.gz
- Extract tar.bz2/bzip archives
  - tar -xvjf archivefile.tar.bz2
- Extract files to a specific directory or path
  - tar -xvzf abc.tar.gz -C /opt/folder/

Creating and Extracting archives (II)

- Extract a single file
  - tar -xz -f archive.tar.gz "./new/file.txt"
- Extract multiple files
  - tar -xv -f abc.tar.gz "./new/cde.txt" "./new/abc.txt“
- Extract multiple files using wildcards
  - tar -xv -f abc.tar.gz --wildcards "*.txt“

Automatically perform tasks – cron (I)

- **cron** is the system process which will automatically perform tasks for you according to a set schedule. The schedule is called the crontab, which is also the name of the program used to edit that schedule.
- The **crontab** is a list of commands that you want to run on a regular schedule, and also the name of the command used to manage that list.

```
MAILTO=[redacted]
#  ┌─────────────────────────── 2. Minute - Minutes after the hour (0-59)
#  │
#  │     ┌──────────────────────── 2. Hour - 24-hour format (0-23).
#  │     │
#  │     │     ┌──────────────────── 3. Day - Day of the month (1-31)
#  │     │     │
#  │     │     │     ┌────────────────── 4. Month - Month of the year (1-12)
#  │     │     │     │
#  │     │     │     │     ┌──────────────5. Weekday - Day of the week. (0-6, where 0 indicates Sun$
#  │     │     │     │     │
#  │     │     │     │     │
#__│_____│_____│_____│____│___Command_____________________________________________________________

# Web stats at https://[redacted]
   1   */1     *     *     *   perl /usr/lib/cgi-bin/awstats.pl -config=web -update >/dev/null
   1   */1     *     *     *   perl /usr/lib/cgi-bin/awstats.pl -config=smtp -update >/dev/null
# Backup LDAP at 03:00
   0     3     *     *     *   bash /var/vmail/backup/backup_openldap.sh
# Backup mysql at 03:30
  30     3     *     *     *   bash /var/vmail/backup/backup_mysql.sh
```

Automatically perform tasks – cron (II)

- **How to use crontab**
  - In BASH issue the following commands:
    - crontab –e // edit the cron for the user you are currently logged in with
    - crontab –l // list the current crontab file
    - The crontab file is usually edited with the **vi** text editor

Automatically perform tasks – backup with tar & cron (I)

- Backup your files with **tar:**
  - tar -cf backup.tar /var/www/vhosts/
  - tar -cvz -f archive-$(date +%Y%m%d).tar.gz /var/www/vhosts/
  - nohup tar -cf backup.tar /var/www/vhosts/ & // this will keep the backup running if you disconnect from the BASH session
  -
  - **Use crontab to schedule automatic backup:**
  - Add this line to crontab to backup your files every day at 4:00 AM
    - 0 4 * * * tar -cvz -f archive-$(date +%Y%m%d).tar.gz /var/www/vhosts/

Automatically perform tasks – backup with tar & cron (II)

- **Use crontab to schedule automatic backup:**
- Add this line to crontab to backup your files every day at 4:00 AM
  - 0 4 * * * /bin/tar -cvz -f archive-$(date +%Y%m%d).tar.gz /var/www/vhosts/

The steps:

1. crontab -e
2. Go to the end of the file
3. Press the “i” key (for insert)
4. Paste the backup command here (push the scroll button on the mouse or shift+insert)
5. Press the ESC key
6. Type :wq //to save and close crontab:

The iptables firewall (I)

What is iptables ?

Iptables is a rule based firewall system and is normally pre-installed on a Linux operating system which is controlling the incoming and outgoing packets. By-default the iptables is running without any rules, we can create, add, edit rules to it.

- service iptables start|stop|restart|status // check the status of the iptables service in Redhat/CentOS
- sudo iptables -L -n -v // check the status of the iptables service in Debian, Ubuntu

The iptables firewall (II)

- Iptables -L // list the current rules of the iptables firewall
- iptables –flush // delete all the rules temporarily.

```
[root@fcsteaua home]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  92.83.92.229         anywhere
DROP       all  --  92.83.92.229         anywhere
DROP       all  --  92.83.92.229         anywhere
DROP       all  --  1.80.0.0/13          anywhere
DROP       all  --  1.92.0.0/14          anywhere
DROP       all  --  192.1.broad.ha.dynamic.163data.com.cn/13  anywhere
DROP       all  --  0.0.202.1.static.bjtelecom.net/15  anywhere
DROP       all  --  1.204.0.0/14         anywhere
DROP       all  --  14.144.0.0/12        anywhere
DROP       all  --  14.208.0.0/12        anywhere
DROP       all  --  23.80.54.0.rdns.as15003.net/24  anywhere
DROP       all  --  23.104.141.0.rdns.as15003.net/24  anywhere
DROP       all  --  23.105.14.0.rdns.racklot.com/24  anywhere
DROP       all  --  27.8.0.0/13          anywhere
DROP       all  --  27.16.0.0/12         anywhere
DROP       all  --  27.36.0.0/14         anywhere
DROP       all  --  27.40.0.0/13         anywhere
DROP       all  --  27.50.128.0/17       anywhere
DROP       all  --  27.54.192.0/18       anywhere
DROP       all  --  27.106.128.0/18      anywhere
DROP       all  --  27.115.0.0/17        anywhere
DROP       all  --  27.148.0.0/14        anywhere
^C
[root@fcsteaua home]#
```

- (Me blocking (not) most of China's IPs)

The logs (I)

- The default log folder in Linux is /var/log
- How do I view log files on Linux?
- Go to /var/log directory using the following cd command:
  - # cd /var/log
- To list files use the following ls command:
  - # ls or ls -lh

```
[root@fcsteaua log]# ls
audit           dracut.log            maillog.processed.1.gz  plesk                    spooler
boot.log        dracut.log-20150501   maillog.processed.2.gz  psa-horde                spooler-20150524
btmp            fcs_cron.log          maillog.processed.3.gz  sa-update.log            spooler-20150531
btmp-20150601   httpd                 mailman                 sa-update.log-20150301   spooler-20150607
ConsoleKit      iptraf                messages                sa-update.log-20150401   spooler-20150614
cron            lastlog               messages-20150524       sa-update.log-20150501   sw-cp-server
cron-20150524   maillog               messages-20150531       sa-update.log-20150601   tallylog
cron-20150531   maillog-20150524      messages-20150607       secure                   wtmp
cron-20150607   maillog-20150531      messages-20150614       secure-20150524          yum.log
cron-20150614   maillog-20150607      mysqld.log              secure-20150531          yum.log-20150101
dmesg           maillog-20150614      newrelic                secure-20150607
dmesg.old       maillog.processed     ntpstats                secure-20150614
[root@fcsteaua log]#
```

The logs (II)

- **Common logs and their location in Linux:**
  - /var/log/messages : General message and system related stuff
  - /var/log/auth.log : Authenication logs
  - /var/log/kern.log : Kernel logs
  - /var/log/cron.log : Crond logs (cron job)
  - /var/log/maillog : Mail server logs
  - /var/log/qmail/ : Qmail log directory (more files inside this directory)
  - /var/log/httpd/ : Apache access and error logs directory
  - /var/log/lighttpd/ : Lighttpd access and error logs directory
  - /var/log/boot.log : System boot log
  - /var/log/mysqld.log : MySQL database server log file
  - /var/log/secure or /var/log/auth.log : Authentication log
  - /var/log/utmp or /var/log/wtmp : Login records file
  - /var/log/yum.log : Yum command log file

The logs (III)

- Display a specific log file:
  - # **less** /var/log/messages
    # **more** -f /var/log/messages
    # **cat** /var/log/messages
    # **tail** -f /var/log/messages
    # **grep** -i error /var/log/messages
  - **grep** with pretty colors:

```
[root@fcsteaua log]# grep -i --color error /var/log/httpd/error_log
[Sun Jun 14 03:36:01 2015] [error] [client 129.130.252.140] client sent HTTP/1.1 request without hostname
 (see RFC2616 section 14.23): /
[Sun Jun 14 15:41:23 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/jmx-console
[Sun Jun 14 15:41:24 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/script
[Sun Jun 14 15:41:24 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/jenkins
[Sun Jun 14 15:41:24 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/hudson
[Sun Jun 14 15:41:24 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/login
[Sun Jun 14 15:41:24 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/jenkins
[Sun Jun 14 15:41:24 2015] [error] [client 89.163.225.224] File does not exist: /var/www/vhosts/default/h
tdocs/hudson
```

The logs (IV) – empty large (log) files

- To empty large files you need to issue one of the following commands:
  - **>** /path/to/large/logfile
  - **echo** “ ” > /path/to/large/logfile

```
[root@fcsteaua ~]# du -h /var/log/messages
1.7M    /var/log/messages
[root@fcsteaua ~]# > /var/log/messages
[root@fcsteaua ~]# du -h /var/log/messages
0       /var/log/messages
[root@fcsteaua ~]#
```

- In the screen shot above I am emptying my 1.7 MB /var/log/messages log file

Networking in Linux (I)

The is no “Local area connection”

Naming convention is:

- eth0
- eth1, etc

Subinterfaces/virtual network cards are noted with “.”

- eth0.1, eth0.2,
- eth 1.1, eth1.2, etc

Networking config files are in /etc/sysconfig/network-scripts/

```
[root@fcsteaua log]# cd /etc/sysconfig/network-scripts/
[root@fcsteaua network-scripts]# ls
ifcfg-eth0   ifdown-ippp  ifdown-routes  ifup-bnep  ifup-plip    ifup-sit          network-functions
ifcfg-lo     ifdown-ipv6  ifdown-sit     ifup-eth   ifup-plusb   ifup-tunnel       network-functions-ipv6
ifdown       ifdown-isdn  ifdown-tunnel  ifup-ippp  ifup-post    ifup-wireless     route-eth0
ifdown-bnep  ifdown-post  ifup           ifup-ipv6  ifup-ppp     init.ipv6-global
ifdown-eth   ifdown-ppp   ifup-aliases   ifup-isdn  ifup-routes  net.hotplug
[root@fcsteaua network-scripts]#
```

Networking in Linux (II)

Modify DNS servers:

- /etc/resolv.conf - is the file you need

List it's contents with

- cat /etc/resolv.conf

```
[root@fcsteaua network-scripts]# cat /etc/resolv.conf
### Hetzner Online AG installimage
# nameserver config
nameserver 213.133.100.100
nameserver 213.133.98.98
nameserver 213.133.99.99
```

- Add or delete existent DNS servers, just edit /etc/resolv.conf with a text editor (vi, nano, etc)

```
  GNU nano 2.0.9                    File: /etc/resolv.conf

### Hetzner Online AG installimage
# nameserver config
nameserver 213.133.100.100
nameserver 213.133.98.98
nameserver 213.133.99.99
nameserver 2a01:4f8:0:a0a1::add:1010
nameserver 2a01:4f8:0:a102::add:9999
nameserver 2a01:4f8:0:a111::add:9898
```

Networking in Linux (III)

ifconfig - ifconfig stands for "interface configuration". It is used to view and change the configuration of the network interfaces on your system. See **ipconfig** in Windows.

```
[root@fcsteaua log]# ifconfig
eth0      Link encap:Ethernet  HWaddr 44:8A:5B:D4:4A:87
          inet addr:136.243.1.30  Bcast:136.243.1.30  Mask:255.255.255.255
          inet6 addr: 2a01:4f8:211:1a9d::2/64 Scope:Global
          inet6 addr: fe80::468a:5bff:fed4:4a87/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:125997981 errors:0 dropped:0 overruns:0 frame:0
          TX packets:156497973 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:44584711776 (41.5 GiB)  TX bytes:163348581085 (152.1 GiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1332127 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1332127 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:329958198 (314.6 MiB)  TX bytes:329958198 (314.6 MiB)

[root@fcsteaua log]#
```

ip address show eth0

Networking in Linux (IV)

netstat – a useful tool for checking your network configuration and activity

```
[root@fcsteaua log]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name

tcp        0      0 136.243.1.30:53         0.0.0.0:*               LISTEN      24339/named

tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      24339/named

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      24226/sshd

tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      24339/named

tcp        0      0 0.0.0.0:8443            0.0.0.0:*               LISTEN      1975/sw-cp-server

tcp        0      0 0.0.0.0:6308            0.0.0.0:*               LISTEN      1975/sw-cp-server

tcp        0      0 0.0.0.0:8880            0.0.0.0:*               LISTEN      1975/sw-cp-server

tcp        0      0 :::465                  :::*                    LISTEN      1505/xinetd
```

Networking in Linux (VI)

route - view and manipulate the TCP/IP routing table in both Unix-like and Microsoft Windows operating systems.

Or **ip route list**

```
[root@fcsteaua log]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
136.243.1.1     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
0.0.0.0         136.243.1.1     0.0.0.0         UG    0      0        0 eth0
[root@fcsteaua log]#
```

```
[root@fcsteaua log]# ip route list
136.243.1.1 dev eth0  proto kernel  scope link  src 136.243.1.30
169.254.0.0/16 dev eth0  scope link  metric 1002
default via 136.243.1.1 dev eth0
[root@fcsteaua log]#
```

Networking in Linux (VII)

Add a default route:

- ip route add default via 192.168.1.254

Delete route from table:

- ip route delete 192.168.1.0/24 dev eth0

Capture and display all packets on interface eth0

- tcpdump -i eth0

Monitor all traffic on port 80 (HTTP)

- tcpdump -i eth0 'port 80'

Networking in Linux (VIII)

ping – utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer and back.

In Windows you need to ping –t to ping forever.

In Linux this is the default behaviour. Ctrl+C or Ctrl+Z to stop any Linux command from running continuous.

Networking in Linux (IX)

ping can be blocked by any firewall software. Is there an alternative to ping ?

Yes.

Introducing **hping. -** hping is a free packet generator and analyser for the TCP/IP protocol.

```
[root@fcsteaua network-scripts]# ping amazon.com
PING amazon.com (176.32.98.166) 56(84) bytes of data.
^C
--- amazon.com ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4525ms

[root@fcsteaua network-scripts]# hping -S -p 80 amazon.com
HPING amazon.com (eth0 176.32.98.166): S set, 40 headers + 0 data bytes
len=46 ip=176.32.98.166 ttl=238 DF id=11560 sport=80 flags=SA seq=0 win=8190 rtt=91.5 ms
len=46 ip=176.32.98.166 ttl=238 DF id=14554 sport=80 flags=SA seq=1 win=8190 rtt=93.9 ms
len=46 ip=176.32.98.166 ttl=239 DF id=44755 sport=80 flags=SA seq=2 win=8190 rtt=97.7 ms
len=46 ip=176.32.98.166 ttl=238 DF id=29169 sport=80 flags=SA seq=3 win=8190 rtt=93.3 ms
^C
--- amazon.com hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 91.5/94.1/97.7 ms
[root@fcsteaua network-scripts]#
```

Networking in Linux (IX)

DNS tools:

host - host is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa.

dig - is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried.

Examples:

- host ericom.com 8.8.8.8
- dig @8.8.8.8 eircom.com in A

Networking in Linux (X)

nmap - (Network Mapper) is a security scanner used to discover hosts and services on a computer network, thus creating a "map" of the network.

What can be done with nmap ?

- Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.
- Port scanning – Enumerating the open ports on target hosts.
- Version detection – Interrogating network services on remote devices to determine application name and version number.
- OS detection – Determining the operating system and hardware characteristics of network devices.

Networking in Linux (XI)

nmap example. Probing for open ports

```
[root@fcsteaua ~]# nmap amazon.com

Starting Nmap 5.51 ( http://nmap.org ) at 2015-06-18 14:36 CEST
Nmap scan report for amazon.com (72.21.206.6)
Host is up (0.099s latency).
Other addresses for amazon.com (not scanned): 205.251.242.103 176.32.98.166
rDNS record for 72.21.206.6: 206-6.amazon.com
Not shown: 994 filtered ports
PORT     STATE  SERVICE
53/tcp   closed domain
80/tcp   open   http
443/tcp  open   https
843/tcp  closed unknown
6881/tcp closed bittorrent-tracker
6969/tcp closed acmsoda

Nmap done: 1 IP address (1 host up) scanned in 7.94 seconds
[root@fcsteaua ~]#
```

Networking in Linux (XII)

nmap example. OS detection and open ports

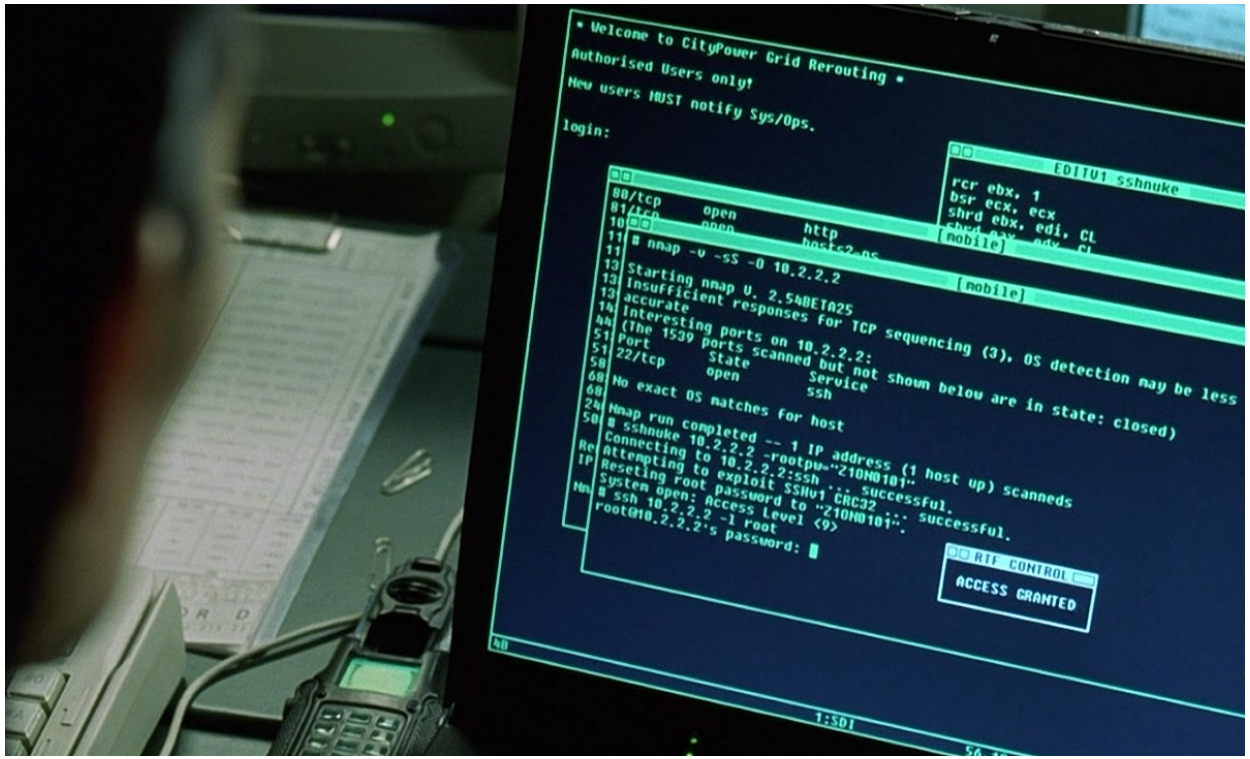
```
[root@fcsteaua ~]# nmap -O wikipedia.com

Starting Nmap 5.51 ( http://nmap.org ) at 2015-06-18 14:42 CEST
Nmap scan report for wikipedia.com (91.198.174.192)
Host is up (0.011s latency).
rDNS record for 91.198.174.192: text-lb.esams.wikimedia.org
Not shown: 995 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
179/tcp  open     bgp
443/tcp  open     https
5666/tcp filtered nrpe
Device type: WAP|specialized|general purpose|PBX|webcam
Running (JUST GUESSING): Netgear embedded (94%), Crestron 2-Series (93%), Linux 2.6.X|2.4.X (89%), V
odavi embedded (88%), AXIS Linux 2.6.X (85%)
Aggressive OS guesses: Netgear DG834G WAP (94%), Crestron XPanel control system (93%), Linux 2.6.22
(89%), Linux 2.6.17 - 2.6.35 (89%), Linux 2.6.23 - 2.6.33 (88%), Vodavi XTS-IP PBX (88%), Linux 2.6.
31 (87%), Linux 2.4.26 (Slackware 10.0.0) (87%), Linux 2.6.24 (87%), Linux 2.6.13 - 2.6.31 (87%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.12 seconds
```

Networking in Linux (XII)

FUN FACT. nmap is so cool that it starred in **The Matrix** movie.




THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
BOLZÏN
MERCI
BÏYAN
SHUKRIA
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
MEHRBANI
GRAZIE
MAAKE
KOMAPSUMNIDA
JUSPAXAR
GOZAIMASHITA
EFCHARISTO
PALDIES
TINGKI
SPASSIBO
SNACHALHUYA
NUHUN
CHALTU
WABEEJA
MAITEKA
YUSPAGARATAM
HUI
GUI
HATUR
UNALCHEESH
SPASIBO
DENKAUJA
NENACHALHYA
ATTO
ANIHA
DHANYADAAD
MERSI
EKOJU
SIKOMO
MAKETAI
MINMONCHAR
SANCO
MERASTAWHY
GAEJTHO
LAH
AGUYJE
FAKAAUE
TAVTAPUCH
MEDAWAGSE
BAINA