

Operating Systems

Practical Courses

By Tom Barbette, based on Sylvain Martin's work

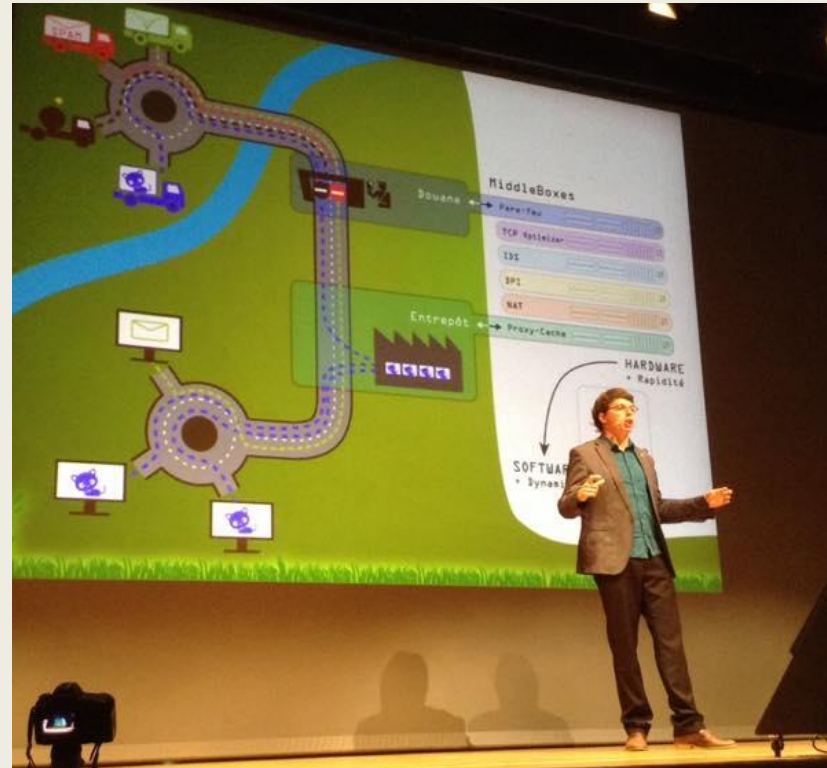
Presentation

Name : Tom Barbette

Mail : tom.barbette@uliege.be

Office : B37 (Math) 1/13

Site : www.tombarbette.be/courses/os/



Summary

- 1. Introduction**
- 2. The computer**
- 3. The Shell**
- 4. Linux Fundamentals**
- 5. Project 1 : Your own shell**

PCs & Projects

1. Well... I just said it
2. Bash & Kernel Interaction
3. Virtual Machines & Kernel Compilation
4. Programming in the Linux Kernel
5. Read vs Mmap
6. Modules
7. File systems

Resources

Resources

- <http://www.tombarbette.be/courses/os/> all infos, links, reference image, posts about general problems that I see in multiple groups and a lot of “tips” that I gave to lost students in the past
 - I’ll add posts from time to time !
- The theory course and books.
- Slides (links on that same page)

- Google, stackoverflow, ...
- <https://www.kernel.org/doc/> <http://www.win.tue.nl/~aeb/linux/lk/lk.html>
- In the kernel sources : .../kernel-4.4.50/Documentation
- Books : <http://idak.gop.edu.tr/esmeray/UnderStandingKernel.pdf>
<http://reiber.org/nxt/pub/Linux/LinuxKernelDevelopment/Linux.Kernel.Development.3rd.Edition.pdf>

- The course discussion forum on the submission platform
- And after a good search... me.



The Computer

What's inside?

- CPU
 - What is it?
 - What does it do?
 - Is there a program loaded in it or something?
- RAM
 - What is it?
- GPU
 - Is it really needed?
- Disk
 - What types?
 - What sizes?
- Expansion cards (PCI, PCIe)
 - Bus? Still actual?

Booting

- BIOS
- Bootloader
 - Is it an OS?
 - Is it a program?
- Kernel
 - Is it the OS?
 - Is it a program?
- Shell
 - Is it a program?
 - What is special with it?
- Is the kernel still running? Where is it?



The Shell

What's that?



The Shell

What's that?

- What you see in movies involving hackers
- Takes commands
- Execute them
- Show you the result

```
Ubuntu 12.04.1 LTS OS-course tty2
OS-course login: student
Password:
Last login: Mon Oct 14 11:08:22 CEST 2013 on tty1
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

 * Documentation:  https://help.ubuntu.com/

System information as of Mon Oct 14 11:13:31 CEST 2013

System load:  0.0                Processes:            61
Usage of /:   70.4% of 1.12GB     Users logged in:     1
Memory usage: 9%                IP address for eth0: 10.0.2.15
Swap usage:  0%                 IP address for virbr0: 192.168.122.1

Graph this data and manage this system at https://landscape.canonical.com/

student@OS-course:~$ _
```

Prompt

The hostname
Your computer name

What you have to type

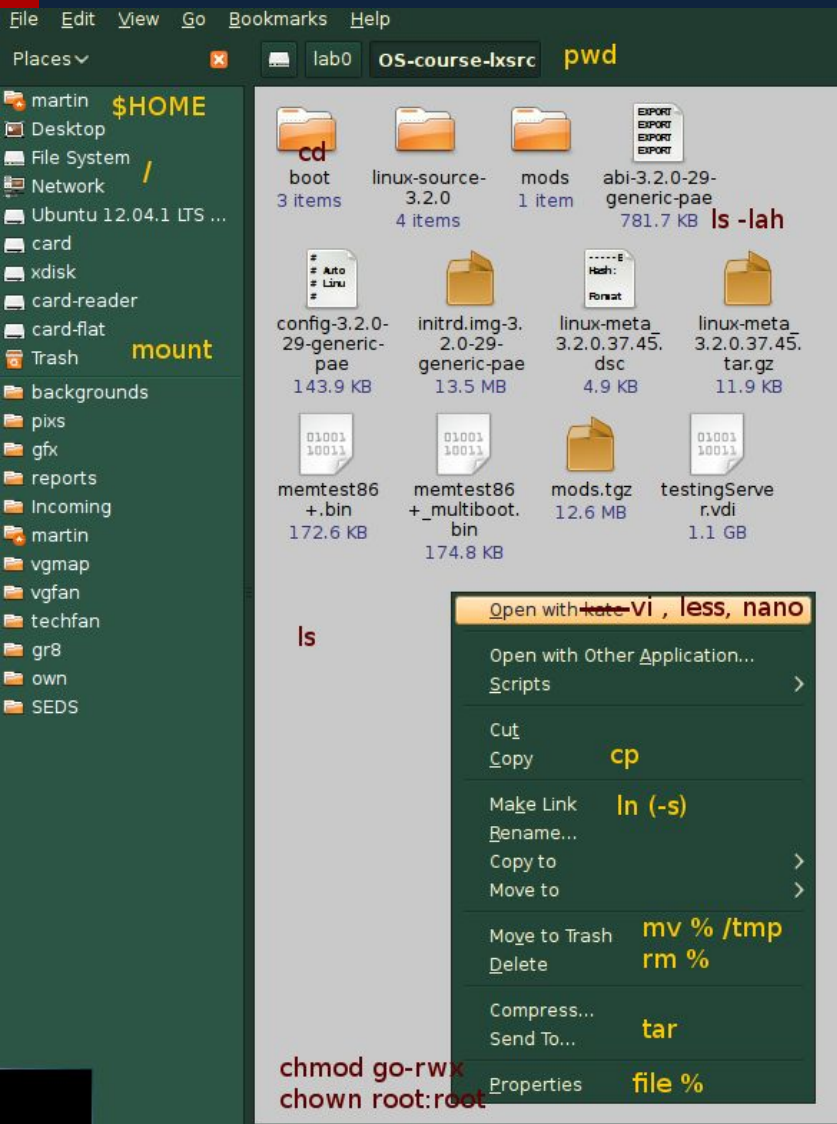
```
host % echo "Hello INFO0940!"
```

```
Hello INFO0940!
```

Terminal

The program output
The result

Basic commands



- type commands, even to *list* (**ls**) and change directory (**cd**)
- every time the system is ready, you see the *prompt*
- most commands have *manual pages*. Try e.g.

```
OS-course:~$ apropos print
```

```
....
```

```
wc (1) - print newline, word ...
```

```
whoami (1) - print
```

```
effective userid
```

```
wprintf (3) - formatted
```

```
wide-char
```

```
OS-course:~$ man wc
```

The real console

- If you are on Linux, try CTRL+MAJ+ALT F2
 - Go back to the “X terminal” with CTRL+ALT F1 or F7 depending on your distribution
- No GUI! You can install gpm to have a very restricted mouse support for copy-paste.

```
Ubuntu 12.04.1 LTS OS-course tty2
OS-course login: student
Password:
Last login: Mon Oct 14 11:08:22 CEST 2013 on tty1
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

* Documentation:  https://help.ubuntu.com/

System information as of Mon Oct 14 11:13:31 CEST 2013

System load:  0.0                Processes:            61
Usage of /:   70.4% of 1.12GB    Users logged in:     1
Memory usage: 9%                IP address for eth0: 10.0.2.15
Swap usage:  0%                IP address for virbr0: 192.168.122.1

Graph this data and manage this system at https://landscape.canonical.com/

student@OS-course:~$ _
```

Working with the console

commands history

```
280 tail -f /var/log
281 bg
282 env
283 envapropos
284 cd
285 apropos print
```

up/down



```
pr (1) - convert text files for printing
print (1) - execute programs via entries in the mailcap file
printenv (1) - print all or part of environment
printerbanner (1) - print large banner on printer
printf (1) - format and print data
printf (3) - formatted output conversion
portstat (1) - print statistics of a process
psinfo (3) - print signal message
psignal (3) - print signal message
pwd (1) - print name of current/working directory
pygmentizer (1) - pretty-print Python source as HTML
python (1) - print python version information
readlink (1) - print value of a symbolic link or canonical file name
regdbdump (1) - parse and print out regulatory rules file
regprep (1) - parse and print out a pattern
seq (1) - print a sequence of numbers
shasum (1) - Print or Check SHA Checksums
snprintf (3) - formatted output conversion
spkac (1) - SPKAC printing and generating utility
sprintf (3) - formatted output conversion
strings (1) - print the strings of printable characters in files.
stty (1) - change and print terminal line settings
swlabel (1) - print or change the label or UUID of a swap area
swprintf (3) - formatted wide-character output conversion
sys_errlist (3) - print a system error message
sys_nerr (3) - print a system error message
tac (1) - concatenate and print files in reverse
tty (1) - print the file name of the terminal connected to standard input
uname (1) - print system information
usb-devices (1) - print USB device details
users (1) - print the user names of users currently logged in to the system
vasprintf (3) - print to allocated string
vdprintf (3) - print to a file descriptor
version (1ssl) - print OpenSSL version information
vfprintf (3) - formatted output conversion
vfprintf (3) - formatted wide-character output conversion
vprintf (3) - formatted output conversion
vsprintf (3) - formatted output conversion
vsnprintf (3) - formatted output conversion
vsnprintf (3) - formatted wide-character output conversion
vsnprintf (3) - formatted wide-character output conversion
wc (1) - print newline, word, and byte counts for each file
whoami (1) - print effective userid
wprintf (3) - formatted wide-character output conversion
xprintf (3) - formatted wide-character output conversion
```

order
of
output

Shift+PgUp/
Shift+PgDn

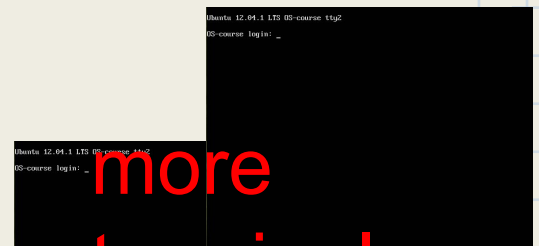
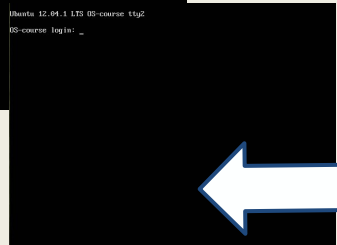
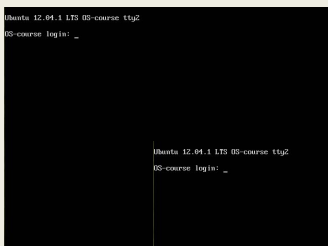
```
shasum (1) - Print or Check SHA Checksums
snprintf (3) - formatted output conversion
spkac (1ssl) - SPKAC printing and generating utility
sprintf (3) - formatted output conversion
strings (1) - print the strings of printable characters in files.
stty (1) - change and print terminal line settings
swlabel (1) - print or change the label or UUID of a swap area
swprintf (3) - formatted wide-character output conversion
sys_errlist (3) - print a system error message
sys_nerr (3) - print a system error message
tac (1) - concatenate and print files in reverse
tty (1) - print the file name of the terminal connected to stand...
uname (1) - print system information
usb-devices (1) - print USB device details
users (1) - print the user names of users currently logged in to t...
vasprintf (3) - print to allocated string
vdprintf (3) - print to a file descriptor
version (1ssl) - print OpenSSL version information
vfprintf (3) - formatted output conversion
vfprintf (3) - formatted wide-character output conversion
vprintf (3) - formatted output conversion
vsprintf (3) - formatted output conversion
vsnprintf (3) - formatted output conversion
vsnprintf (3) - formatted wide-character output conversion
vsnprintf (3) - formatted wide-character output conversion
wc (1) - print newline, word, and byte counts for each file
whoami (1) - print effective userid
wprintf (3) - formatted wide-character output conversion
xprintf (3) - formatted wide-character output conversion
student@OS-course:~/mp$ cd
student@OS-course:~$ apropos print
```

order
of
output

more
terminals

Alt+=

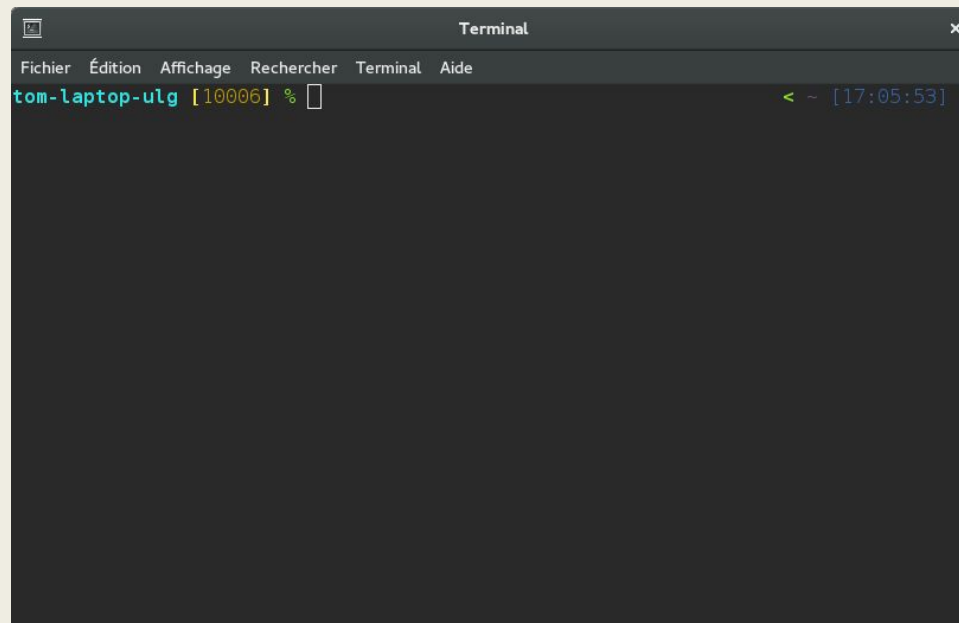
Alt+=



Pseudo Terminal

Painless terminal

- All Linux and Mac OS X come with a pseudo-terminal
- You can open it like any other application
- Easy selection, copy-paste from web, go back in history using scroller, ...

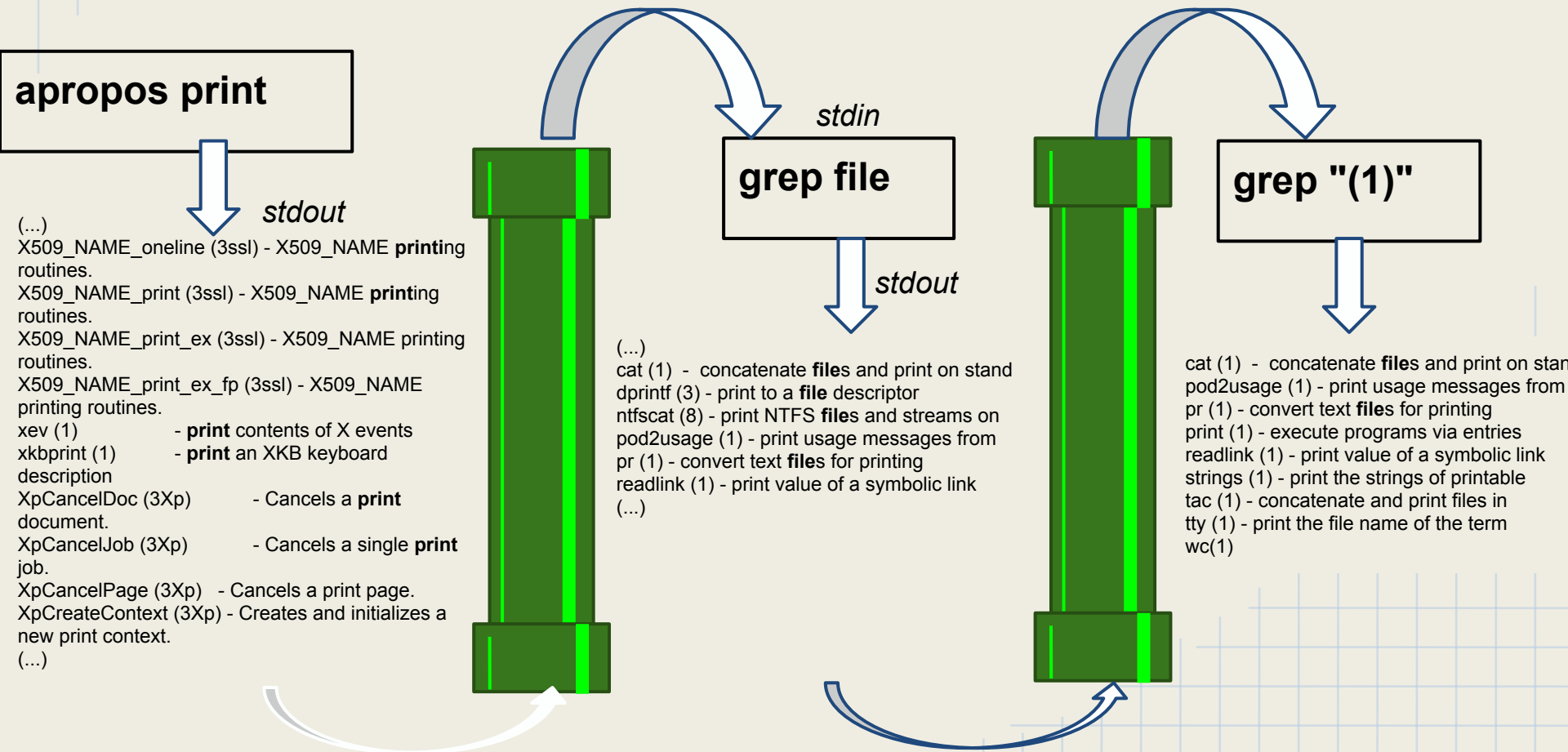


Chaining | commands

```
$ apropos print | grep file | grep (1)
```

Terminal

3 process created, output chained to inputs using so-called *pipes*



One shell to rule them all?

- By default, it is bash nearly everywhere
- **ZSH** (sudo apt-get install zsh && chsh) is just better
 - More intelligent autocompletion with TAB
 - E.g type “ls /home/t” and TAB will propose “ls /home/tom”
 - I said, just better !
- You can personalize all, changing your configuration in ~/.bashrc for bash and ~/.zshrc for ZSH
- An older version of mine at <http://queen.run.montefiore.ulg.ac.be/~barbette/tag/zshrc/>
- At least add some aliases in your rc file :
 - alias i="sudo apt-get install" → “i program” to install program instead of typing the whole “sudo apt-get install program” every time
- **You can save a lot of time !**



Linux fundamentals

Linux

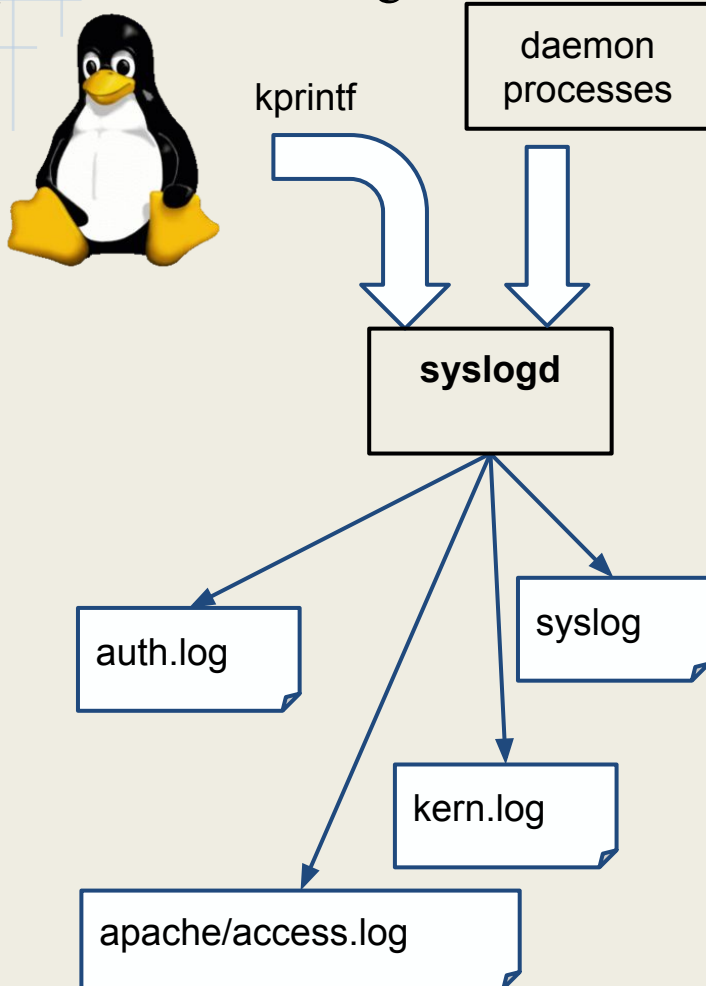
- Is the name of the ***Kernel***
 - Made by Linus Torvalds
- By extension, also the name of the *distributions* that uses it
 - In general Linux distributions put binaries in the /bin folder, and programs in /usr, though the Linux kernel has nothing to do with that
- Today : Linux-based Operating Systems general ideas
- Next week : Interactions with the Linux Kernel itself

Folders

- /bin
 - Software binaries (system/basic ones)
 - bash, ls, zip, tar , ...
- /sbin
 - Super-user binaries
 - fdisk, gparted, ...
- /dev, /proc, /sys access to kernel interaction (next week)
- /home
 - Users files
- /usr
 - Program installed by users
- /mnt, /media
 - Mount points (USB keys, network, ...)
- /var
 - Software data (web root, caches, SQL database, ...)
- /etc
 - Configuration files

/var/log

A place where the logs are collected




- **less /var/log/syslog**
navigate through past events (/ to enter a regexp to search, n to hop to the next match, q to quit)
- **tail -f /var/log/auth.log &**
will report bad users.

? *Can you tell when the system was last re-booted ? Which file was used as boot_image ?*

sudo vi /etc/passwd

- **/etc** stores all the machine's config
- (mostly as text files)
- **sudo** allows some users to *act as system administrator* (root) for *one command*.
- If you prefer, **nano** is installed in the Virtual Machine we'll use.

 Can you change the machine's hostname in the configuration ?

Installing programs

- **apt-cache search “XXX”**
- **sudo apt-get install XXX**

```
student $ apt-cache search jed
```

```
(...)
```

```
jed - Éditeur pour programmeur
```

```
jed-common - Fichiers d'exécution S-lang compilés pour jed et xjed
```

```
slang-gdbm - l'accès aux bases de données GDBM depuis S-Lang
```

```
xjed - éditeur pour programmeurs (version x11)
```

```
student $ sudo apt-get install jed
```

```
Lecture des listes de paquets... Fait
```

```
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
```

```
(...)
```

```
0 mis à jour, 4 nouvellement installés, 0 à enlever et 83 non mis à jour.
```

```
Il est nécessaire de prendre 821 ko dans les archives.
```

```
Après cette opération, 3.064 ko d'espace disque supplémentaires seront utilisés.
```

```
Souhaitez-vous continuer [O/n] ? o
```

```
(...)
```

Terminal

User vs Root

User privilege level
% or \$: user mode
: root mode

```
host % echo "Hello INFO0940!"  
Hello INFO0940!
```

Terminal

WARNING

No link with the CPU privilege mode, SYSCALL mechanism, ... Root is still in “user” context and is absolutely not tied to kernel mode !


Root

Root = Super User = SU

- The root user has access to all files
- In Unix, everything is a file
 - Your Terminal is `/dev/tty`
 - Your USB device is `/dev/usb*`
 - Kernel parameters are in `/proc`
 - ...

→ Root has access to everything in Unix...

(A good *sylogisme* example)

 *How to become root?*

Become the master

```
% make sandwich  
You have to be root !
```

Terminal

First method with su

```
% su  
Password : [Type root password]  
# make sandwich  
Your sandwich is ready !  
#
```

Terminal

Second method with sudo :

```
% sudo make sandwich  
Password : [Type your user password]  
Your sandwich is ready !  
%
```

Preferred !

Terminal

sudo -s allows you to mimic “su” but giving your own password (like sudo su)

Linux fundamentals

Really?

Most of these commands are distribution-specific

- “apt-get” is for debian-based distributions, like Ubuntu. You’ll use “yum” in fedora or red-hat

- /var/log/syslog is for debian-based, but it’s /var/log/messages in redhat-based...

- ...

We’ll use **Ubuntu** for its popularity



Project 1

Deadline 21/02 23:59



Project #1

Implement a simple shell in C

- You will implement a simple shell, which uses exclusively **execv** and **fork** for sub-process.
 - Fork duplicates the process
 - Execv replace the current process by another one
- It must support launching programs with arguments, and return to the prompt when the program terminates, but no other concepts such as variable, substitution, pipes, chaining, ...
- It will exit upon typing "exit" or CTRL+D (EOF)
- Command line can be limited to 255 arguments, while each arguments may be limited to 255 characters.
- The first prompt must be "> ", then "RET> " where RET is the return code of the last command. If there was no command when pressing enter, "> " is shown

Example

```
$ gcc -std=gnu99 -o shell shell.c && ./shell
> /bin/ls
shell shell.c  shell.c~ shell.tar.gz

0> /bin/ls -al
total 32
drwxr-xr-x. 2 tom tom 4096  9 fév 12:23 .
drwxr-xr-x. 3 tom tom 4096  9 fév 11:28 ..
-rwxr-xr-x. 1 tom tom 9032  9 fév 12:23 shell
-rw-r--r--. 1 tom tom 1133  9 fév 12:23 shell.c
-rw-r--r--. 1 tom tom 1134  9 fév 12:23 shell.c~
-rw-r--r--. 1 tom tom  208  9 fév 11:41 shell.tar.gz

0>
```

Terminal

PATH support

- Look in the folders of \$PATH (separated by :) in order to find the program to launch
- This should avoid to type `"/bin/lis"`, as bin should be in the path, typing `lis` should be enough

DO NOT USE `exec*p` variants that do that themselves, only support it manually

/bin/chsh

Will you dare trust your own shell?

Submission

- You must submit a file named `shell.c` using the submission platform (<https://submit.montefiore.ulg.ac.be>)
- Currently, only compile test and a few input/output comparisons
- **Per group of 2** that you will keep the whole semester

Bring your laptop next week !

- Please have a Linux ready for next week with either :
 - Linux (maybe in dual boot)
 - Windows/Mac OS with a Linux in VirtualBox
 - Windows/Mac OS connecting to ms8xx machines using Putty/SSH
 - <http://www.student.montefiore.ulg.ac.be/accounts.php>
 - Please arrange that before the course !
- Do not hesitate to e-mail me

Putty (SSH on Windows) :

<http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Windows 10 Bash is also a good solution for SSH to ms8xx :

<http://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/>