

# Introduction au terminal, SSH et quelques outils.

DABY-SEESARAM ARNAUD

October 20, 2021

- 1 Terminal
- 2 Shells
- 3 SSH
- 4 IRC
- 5 Un petit TP ☺

# Table of contents

- 1 Terminal
- 2 Shells
- 3 SSH
- 4 IRC
- 5 Un petit TP 😊

# Historical terminals



Figure: Terminal of a DEC VT100



# Table of contents

- 1 Terminal
- 2 Shells**
- 3 SSH
- 4 IRC
- 5 Un petit TP 😊

# Shells

## What is a shell ?

The shell is an interactive program that:

- interprets the user input
- performs actions or computes
- can output stuff

Example: `echo -e "Color \033[31mred\033[0m."`

# Example of shells

POSIX-compliant	others
<b>bash</b>	nushell
Zsh	C shell
Dash	fish
yash	ion
...	...

From now on, I will use bash (config file in `~/.bashrc`).



# Basic usage

## Adresses of files

Element	Meaning
/	root of the filesystem
/home/arnaud/dir/file	Path to a file called "file"
/home/arnaud/dir	Path to a directory called "dir"

# Basic usage

## Permissions on files

- Users have names and belong to groups of users
- Every directory and file belongs to a user and a group
- A permission is a triplet:

user	group	other	
------	-------	-------	--

rwX	rwX	rwX	Every permissions for everyone
-----	-----	-----	--------------------------------

rw-	---	---	The owner can read and write.
-----	-----	-----	-------------------------------

r-x	---	---	The owner can read and execute.
-----	-----	-----	---------------------------------

# Basic syntax

The bash syntax support:

- The juxtaposition of commands
- for loops
- while loops
- Conditions
- Variables
- Redirections

# Basic syntax

The bash syntax support:

- The juxtaposition of commands
- for loops
- while loops
- Conditions
- Variables
- Redirections

## Example

Command

---

<code>printf a &amp;&amp; printf b</code>	→	prints "ab"
<code>make &amp;&amp; make test    search_errors</code>	→	...
<code>firefox</code>	→	launches Firefox

# Interprétation

Pour bash, tout est texte.

Il y a plusieurs étapes dans l'interprétations:

- Recherche de wildcards (`*.jpg` se développe en la liste des fichiers d'extension `.jpg`)
- Recherche de substitutions
- Recherche de sous-expressions (`$(...)`)
- Recherche de sous-expressions arithmétiques (`$((...))`)
- Recherche de variables à remplacer (si `a` vaut `2`, `echo $a` est transformé en `echo 2` avant l'affichage)
- Recherche d'accolades à développer (`a{b,e{c,d}}` se transforme en `ab aec aed`)

# Examples

```
make && make test || make search_errors  
for f in *.tmp_file; do rm $f; done  
↔rm *.tmp_files
```

# Table of contents

- 1 Terminal
- 2 Shells
- 3 SSH**
- 4 IRC
- 5 Un petit TP 😊

## Main command

- `ssh user@server` allows you to connect to server as user
- `ssh -J tata user@server` allows you to connect to server as user from a connection to tata
- `ssh-copy-id -i <key> user@server` allows you to give your public key to a server
- `ssh -i <file> user@server` allows you to connect to server as user using the key <file>



## Config file

Location: `~/.ssh/config` to store the host-related information.

Host crans:

```
Hostname zamok.crans.org
```

Example:

```
User ds-ac
```

```
Port 22
```

```
IdentityFile [PRIVATE KEY]
```

```
ProxyJump tata
```

## File sharing

`scp` = `cp` via SSH (e.g. `scp local host:distant`).

Alternative: `rsync`

# Principle

SSH is a protocol that ensures a secure connection between two machines.

Most common usage:

- Open a remote shell
- Execute a single command on a remote server
- Synchronize / Copy files
- Forward a port (port tunneling)
- Forward graphical windows (X forwarding (requires an X server))

# Principle

## Authentication

- Mainly managed with the configuration option

`AuthenticationMethods`

Example `AuthenticationMethods publickey,password`  
forces the user to connect with a valid public key and then a password

- Command restriction on the server.

Example: GitHub: cannot log into a shell

# Table of contents

- 1 Terminal
- 2 Shells
- 3 SSH
- 4 IRC**
- 5 Un petit TP 😊

# Principle

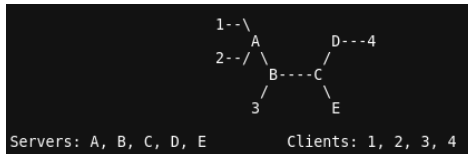


Figure: Graph of an IRC network (source: RCF 1459)

# Most common client commands

```
/connect <server> [<port>]  
/join <channel>[,<channel'[,...]]  
/part <channel>  
/privmsg <dest> <msg>
```

# Tmux

Présentation en direct ☺ !

# Table of contents

- 1 Terminal
- 2 Shells
- 3 SSH
- 4 IRC
- 5 Un petit TP 😊**



## TP

## Commands to interact with files

<code>ls</code>	prints the content of the current directory
<code>cd</code>	change directory
<code>cat</code>	dumps a file content
<code>cp &lt;src&gt; &lt;dest&gt;</code>	copy the source into the destination
<code>mv &lt;src&gt; &lt;dest&gt;</code>	copy the source into the destination
<code>rm &lt;file&gt;</code>	removes a file (permanent!)
<code>touch &lt;file&gt;</code>	create a file or update its timestamps
<code>mkdir &lt;dir&gt;</code>	create a new directory
<code>&lt;cmd&gt; &gt; &lt;file&gt;</code>	dump the output of a command in the file
<code>nano/vim &lt;file&gt;</code>	edit a file using the nano (or the vim) editor

Pour en savoir plus sur une commande, essayer `apropos <cmd>`

## TP

## SSH

- Connect to `zamok.crans.org`
- Launch the `tmux` command

## IRC

- In the `tmux`, open an IRC client and connect to the server `irc.crans.org`
- Set your nickname using `/nick <nick>`
- Join the channel `#seminaire-shell`

## TP

## Shell

- Open a new tmux window
- Generate a random number between 12 and 59 (note that \$RANDOM is a special bash variable that contains a big random number and that % represents the modulus operation).
- Execute `echo !! ; !!` What happens ? What does !! represent ?
- Send the result of the last command in the IRC channel
- Generate a random number between 9 and 16. (hint: You can press `Ctrl-R` to search backward in history)

## TP

## Scripting

Write a script using vim, nano or ed that:

- Create a new directory
- Goes into that directory
- Create a file with touch
- Change its rights so that noone can access it but you
- Edit it as you want
- Sort its lines and filter them. Put the result in another file