Sujet 103 : Commandes GNU et Unix

- 103.1 Travailler en ligne de commande (Weight 4)
- 103.2 Contrôler des flux de texte à l'aide des filtres (Weight 3)
- 103.3 Effectuer la gestion de base des fichiers (Weight 4)
- 103.4 Utilisation des flux, des tubes (pipes) et des redirections (Weight 5)
- 103.5 Création, surveillance et destruction de processus (Weight 5)
- 103.6 Modifier la priorité d'exécution d'un processus (Weight 3)
- 103.7 Recherche sur des fichiers texte avec des expressions régulières (Weight 2)
- 103.8 Édition de fichiers texte avec "vi" (Weight 3)

Travailler en ligne de commande

Travail en ligne de commande

- **Description** : Les candidats doivent être capables de travailler en ligne de commande. l'utilisation du shell bash sera traité dans cet objectif.
- **Termes, fichiers et utilitaires utilisés :**
  - bash
  - echo
  - env
  - exec
  - export
  - pwd
  - set
  - unset
  - uname
  - history

bash

- **Bourne-again shell** compatible avec sh, avec des fonctionnalités de ksh, csh
- un shell est un programme qui exécute des programmes et permet aussi de construire d'autres programmes appelés **scripts.**
- **prompts**

```
[db2inst1@echidna db2inst1]$
ian@lyrebird:~>
$
```

```
[root@echidna root]#
lyrebird:~ #
#
```

salah@Bagdad:~> echo $PS1

\u@\h:\w>

salah@Bagdad:~> echo $PS2

>

Séquences de commandes

- Exécuter séquentiellement des commandes l'une après l'autre :
  - **cmd1 ; cmd2**
- Exécuter cmd2 si et seulement si cmd1 s'est exécutée sans erreur
  - **cmd1 && cmd2**
- Exécuter cmd2 si et seulement si cmd1 a renvoyé une erreur :
  - **cmd1 || cmd2**
- **&** en fin de commande permet de lancer cette commande en tâche de fond (background)
  - **firefox &**

Variables

- **Variables d'environnement** : connues de toutes les commandes les commandes lancées depuis le shell.

```
salah@Bagdad:~> PS1="[\t][\u]\$"
[10:12:32][salah]$
```

- **variable simple** :

```
$ formation="lpi"
$ echo $formation
lpi
```

- Rendre la variable visible pour les shells fils :

```
$ export formation
```

- Afficher toutes les variables d'environnements : **$ env**
- Afficher les variables simples et les variables d'environnement : **$ set**
- Effacer une variable : **$ unset formation**

Quelques variables d'environnement bash

Variable	Fonction
USER	le nom de l'utilisateur courant
UID	UID de l'utilisateur courant
HOME	Le repertoire de connexion de l'utilisateur courant
?	Le code d'erreur de la dernière commande
#	Le nombre de paramètres à l'appel du script
1,2....	les paramètres du script
0	le nom du script

quotes et variables

- **Quote double :** Permet la substitution des variables

  $ echo "mon repertoire est $HOME"

  mon repertoire de connexion est /home/salah

- **Quote simple :** Désactive l'interprétation des caractères spéciaux

  $ echo 'mon repertoire de connexion est $HOME'

  mon repertoire est $HOME

- **Quotes inversées :** Permet la substitution des commandes

  $ echo "mon repertoire courant est `pwd`"

  mon repertoire courant est /tmp

Raccourcies claviers

- history
- HISTSIZE
- HISTFILE

Designator	Description
`!!`	Spoken as bang-bang, this command refers to the most recent command. The exclamation point is often called bang on Linux and Unix systems.
`!n`	Refer to command `n` from the history. Use the history command to display these numbers.
`!-n`	Refer to the current command minus `n` from the history.
`!string`	Refer to the most recent command starting with `string`.
`!?string`	Refer to the most recent command containing `string`.
`^string1^string2`	Quick substitution . Repeat the last command, replacing the first occurrence of `string1` with `string2`.

Raccourcies claviers, commandes Emacs

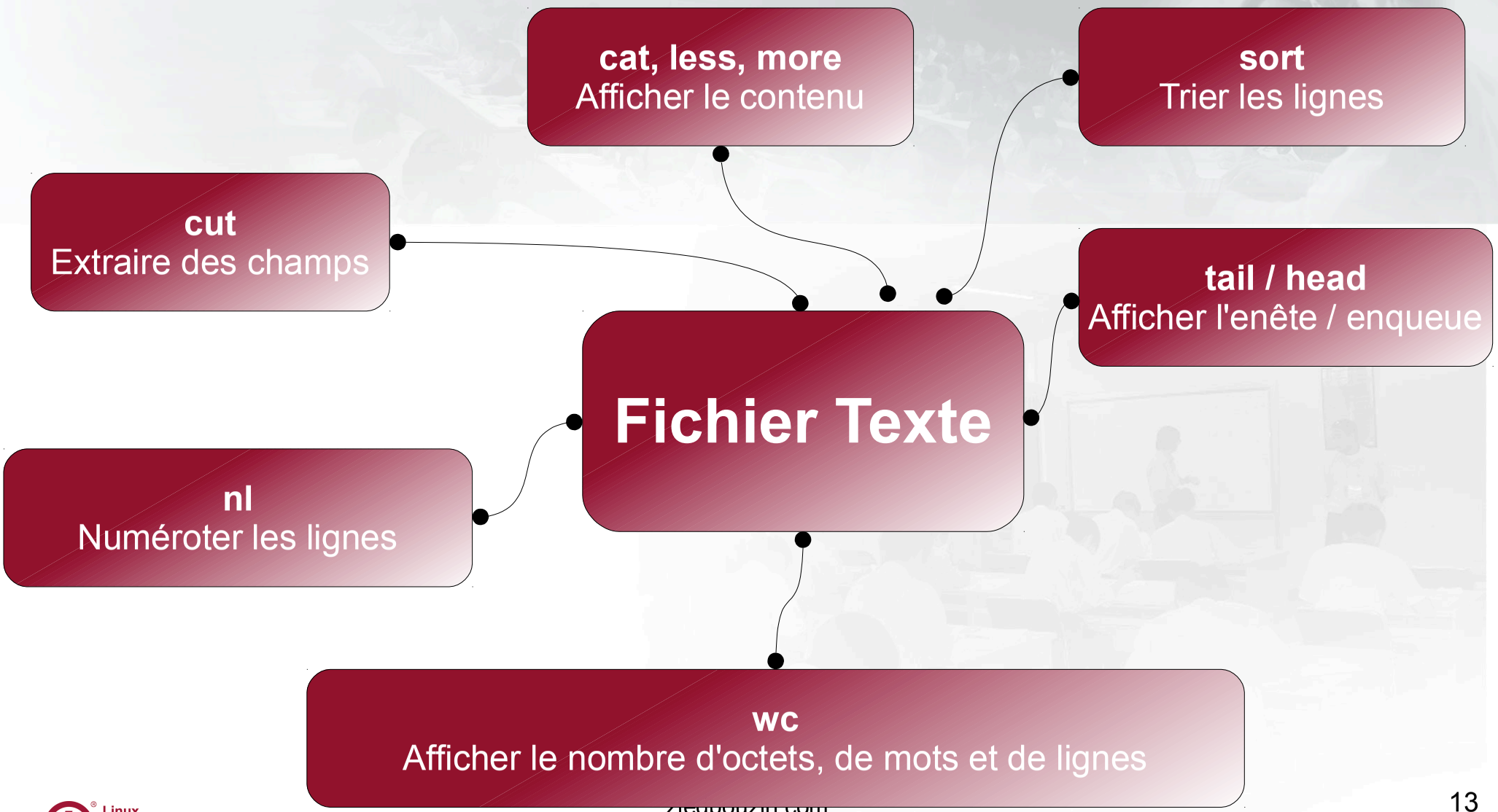
Key	Description
C-p	Previous line (also up arrow)
C-n	Next line (also down arrow)
C-b	Back one character (also left arrow)
C-f	Forward one character (also right arrow)
C-a	Beginning of line
C-e	End of line
C-l	Clear the screen, leaving the current line at the top of the screen
M-<	Top of history
M->	Bottom of history
C-d	Delete character from right
C-k	Delete (kill) text from cursor to end of line
C-y	Paste (yank) text previously cut (killed)
M-d	Delete (kill) word
C-rtext	Reverse search for text
C-stext	Forward search for text

Contrôler des flux de texte à l'aide des filtres

Traitement de flux de type texte par des filtres

- **Description** : Les candidats doivent être capables d'appliquer des filtres à un flux de type texte.
- **Termes, fichiers et utilitaires utilisés :**
  - cat
  - cut
  - expand
  - fmt
  - head
  - od
  - join
  - nl
  - paste
  - pr
  - sed
  - sort
  - split
  - tail
  - tr
  - unexpand
  - uniq
  - wc

Filtres (1)


cat, less, more
Afficher le contenu
sort
Trier les lignes
cut
Extraire des champs
tail / head
Afficher l'enête / enqueue
Fichier Texte
nl
Numéroter les lignes
wc
Afficher le nombre d'octets, de mots et de lignes

cat et tac

- Affiche le contenu d'un fichier.
- **Exemple** fichier1

  1 un

  2 deux

  3 trois

$ cat fichier1

$ tac fichier1

head et tail

- **head** : Afficher le début d'un fichier (par défaut les 10 premiers lignes)
- **Exemple**

  $ head -3 /var/log/messages
- **tail** : Afficher la dernière partie d'un fichier (par défaut les 10 derniers lignes)
- **Exemples :**

  $ tail -c20 /etc/passwd

  $ tail -f /var/log/messages

cut

- Permet d'afficher certains champs d'un fichier donné
- Exemple :

```
$ cut -d: -f1 /etc/passwd
```

nl

- Numéroter les lignes d'un fichier.
- Exemples :

```
$ nl /etc/passwd
$ ls | nl -s')'
```

sort

- Trier les lignes d'un fichier texte
- **options**
  - **-d** : Trier dans l'ordre alphabétique
  - **-n** : Trier dans l'ordre numérique
  - **-r** : Inverser l'ordre
- **Exemples :**

  $ sort /etc/passwd

  – **trier selon le champs RSS (resident size)**

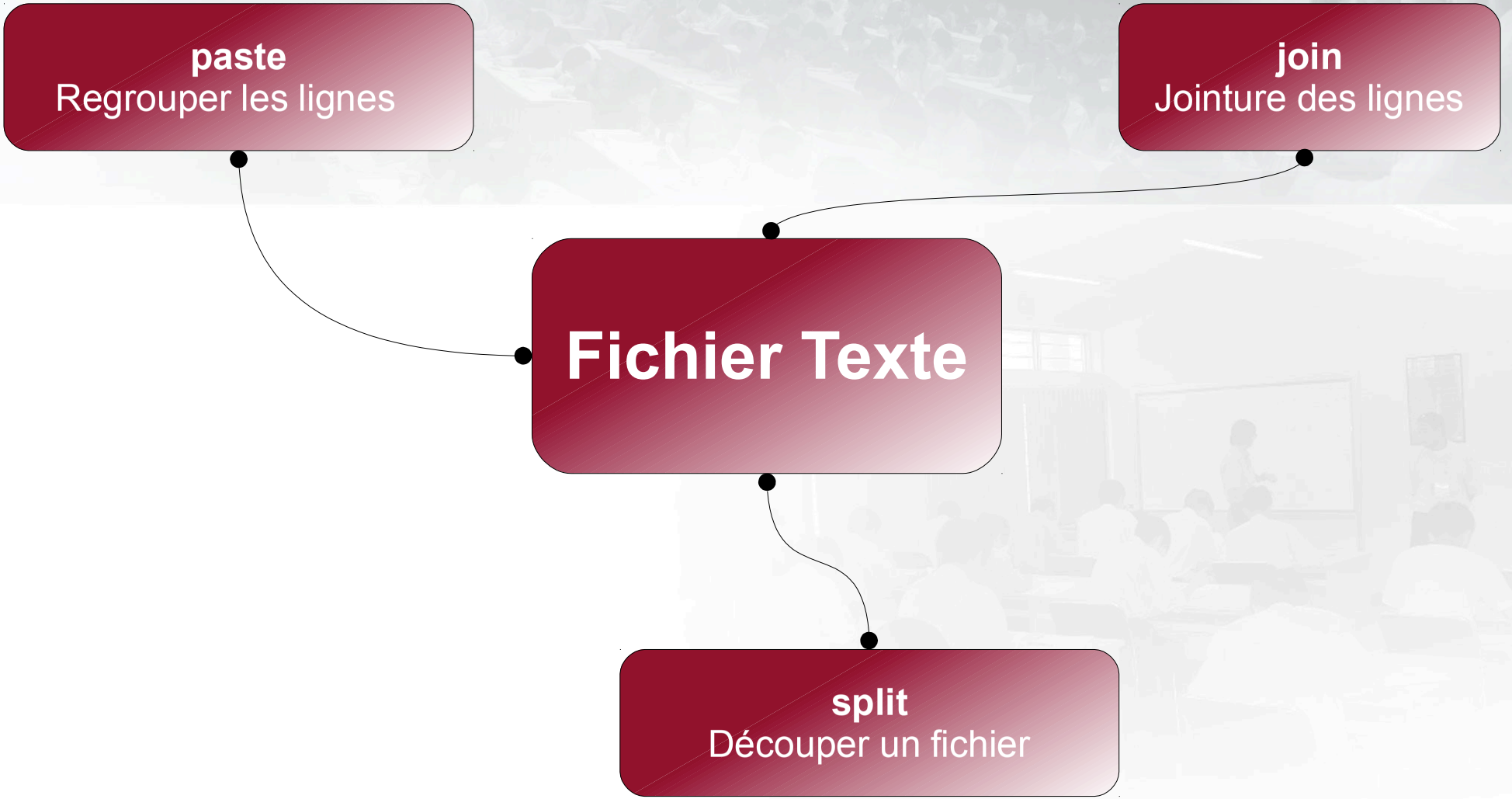
  $ ps aux | sort -k 6 -n

wc

- Afficher le nombre d'octets, de mots et de lignes d'un fichier.
- **Options**
  - **-c :** Afficher uniquement le nombre d'octets
  - **-m** : Afficher uniquement le nombre de caractères
  - **-l** : Afficher uniquement le nombre de lignes
  - **-w** : Affiche uniquement le nombre de mots
- **Exemples :**

```
$ wc -l fich
$ wc -w fich
$wc fich
```

Filtres (2)


paste
Regrouper les lignes
join
Jointure des lignes
Fichier Texte
split
Découper un fichier

paste

- Regrouper les lignes de différents fichiers.
- Exemple :
  - file1

    1

    2

    3
  - file2

    A

    B

    C

```
$ paste file1 file2
1    A
2    B
3    C
```

paste (suite)

```
$ paste -d'@' file1 file2
    1@A
    2@B
    3@C

$ paste -s file1 file2
    1    2    3
    A    B    C
```

join

- Fusionner les lignes de deux fichiers ayant un champ commun.
- **Exemple** :
  - file1 :
    1 one
    2 two
    3 three
  - file2 :
    1 11
    2 22
    3 33
  - $ join -j 1 file1 file2
    1 one 11
    2 two 22
    3 three 33

split

- Découper un fichier en différentes partie.
- **Exemple1** :
  - file1 :

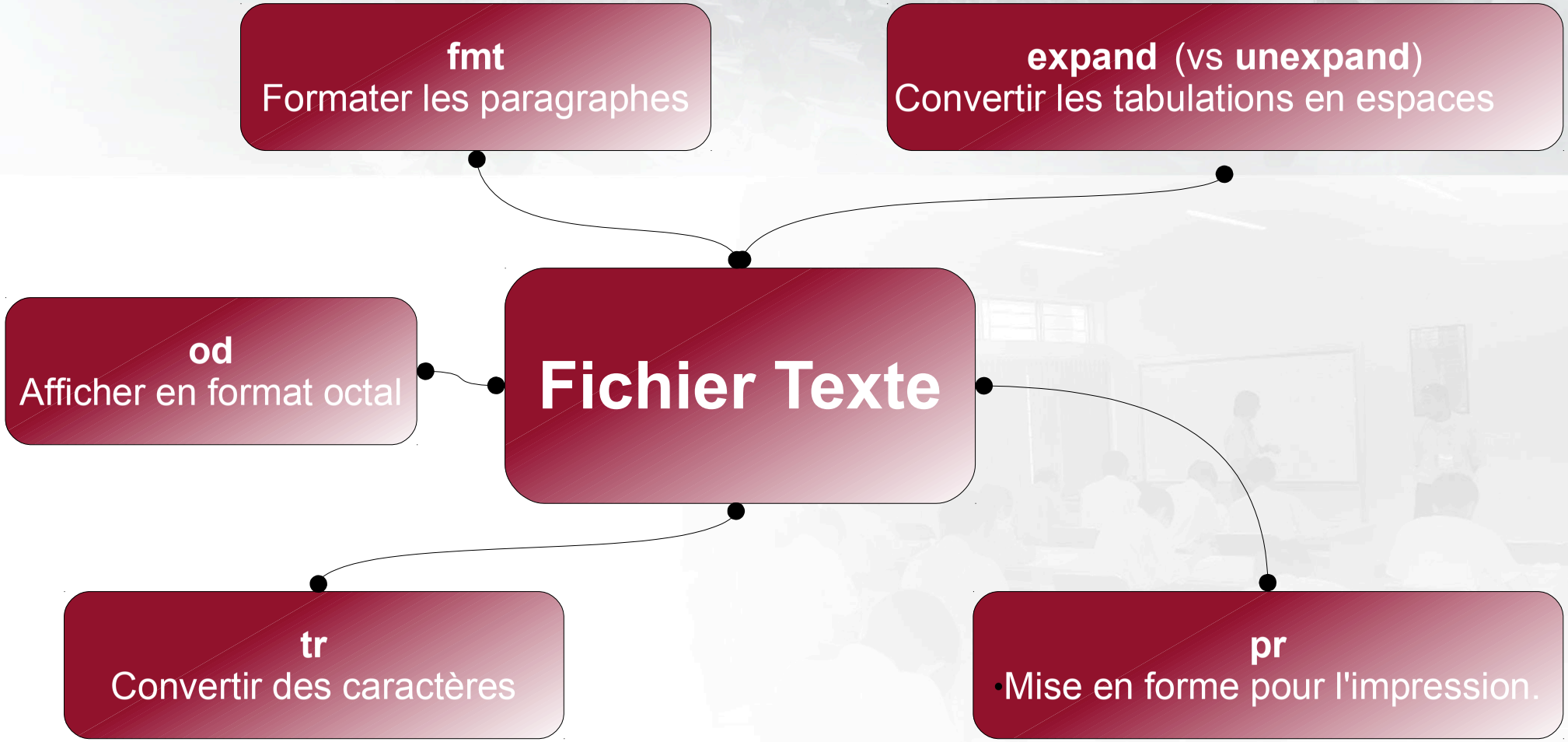
    1 one
    2 two
    3 three
    4 four
    5 five
    6 six

  $ split -2 file1 splitout_

  ==> créer trois fichiers **splitout_aa**, **splitout_ab**, et **splitout_ac**
- **Exemple 2** : Découper un fichier en plusieurs de **taille maximale** d'une disquette :

  $ split -b 1.4m grosfichier petitfichier_

Filtres (3)


fmt
Formater les paragraphes
expand (vs unexpand)
Convertir les tabulations en espaces
od
Afficher en format octal
Fichier Texte
tr
Convertir des caractères
pr
Mise en forme pour l'impression.

expand

- Convertir les tabulations d'un fichier en espaces.
- **unexpand** fait le processus inverse.
- Exemple : convertir les tabulations en 2 espaces:

```
$ expand -t 2 monfichier
```

fmt

- Formater les paragraphes dans un fichier.
- **options** :
  - **-u** : Espacement uniforme. Réduire les espacements entre les mots à une espace
  - **-w** : Remplir les lignes jusqu'à la largeur mentionnée (par défaut 75 colonnes)
- **Exemple** :

```
$ fmt -w 80 myfile.txt > myfile80wide.txt
```

od

- Afficher le contenu d'un fichier en octal ou sous d'autres formats (decimal, hexadecimal, ASCII)
- **option** :

  -t type Sélectionner le format d'affichage des résultats selon le type :

  - **a** : caractères littéraux
  - **c** : caractères ASCII ou séquences d'échappement préfixées par BackSlash
  - **o** : valeurs octales
  - **u** : valeurs décimales non signées
  - **x** : valeurs hexadécimales
- **Exemples** :

```
$ od -t a file1
$od -t c file1
$ od -t x1 file1
```

pr

- Mettre en forme des fichiers de texte pour l'impression.
- **Quelques options :**

  -wN : Indique le nombre de caractères par lignes (72)

  -lN : Indique le nombre de lignes par page (66).

  -hTexte : le Texte doit remplacer le nom de fichier dans l'en-tête de chaque page.

  -n : numérote les ligne

  -m : imprimer tous les fichiers en parallèle un par colonne

- ***Exemples***

  $ pr -hLPI rapport.txt | lp

  $ pr -m budget.oct budget.nov | lp

tr

- Pour effectuer des conversions de caractères (exp minuscule/majuscule, ...)
- **Exemples :**
  - **$ cat file1 | tr 'a-z' 'A-Z' OU $ cat file1 | tr '[:lower:]' '[:upper:]'**
  - la suppression des accents d'un texte :
  - **cat file1 | tr 'àçéèêëîïôöùüÂÇÉÈÊËÎÏÔÖÙÜ' 'aceeeeiioouuACEEEEIIOOUU'**
  - Convertir les séquences de sauts de lignes en un seul saut de ligne (ceci supprime les lignes blanches) :
    - **cat file1 | tr -s '\n'**

Gestion de fichiers

Effectuer une gestion de base sur les fichiers.

- **Description** : Les candidats doivent être capables d'utiliser les commandes Linux de base pour gérer les fichiers et les répertoires.
- **Termes, fichiers et utilitaires utilisés :**
  - cp
  - find
  - mkdir
  - mv
  - ls
  - rm
  - rmdir
  - touch
  - tar
  - cpio
  - dd
  - file
  - gzip
  - gunzip
  - bzip2

Inodes

- Associer à chaque objet du système de fichier un inode (**The identification information for filesystem object**)
- Un inode regroupe des informations sur l'objet du système de fichiers : localisation, date de modification, paramètres de sécurité....
- Chaque système de fichier ext2 est crée avec un nombre fini d'inodes calculé selon la taille du système de fichier et d'autres options passées à la commande **mke2fs**.
- plusieurs objets du système de fichiers peuvent partager le même inode : **lien physique**

Commandes de base

- **pwd** : Afficher le chemin absolu du répertoire courant
- **cd** : changer de répertoire
- **ls** : lister le contenu d'un répertoire
- **mkdir** : créer un nouveau répertoire
- **rmdir** : supprimer un répertoire
- **touch** : changer les informations de date et de d'heure d'un fichier; créer un fichier vide lorsque le fichier passé en argument n'existe pas.
- **cp** :copier un fichier
- **mv** : Déplacer ou renommer un fichier
- **rm** : supprimer un fichier.

Utiliser les wildcards

- Besoin de manipuler plusieurs fichiers à la fois,
- Par exemple : cas d'un développement en C et pour être sure de recompiler tous les fichiers sources.
-

Wildcards	Descriptions	Exemples
*	désigne 0 ou plusieurs caractères	x*
?	Désigne exactement un seul caractère	x? x??
[caractères]	Désigne un seul caractère de la liste [caractères]	x[yz]
[!caractères]	Désigne un seul caractère en dehors de la liste [caractères]	x[!yz]
[a-z]	Désigne un seul caractère appartenant à l'intervalle de caractères défini entre [ ]	x[0-9] x[a-zA-Z]
[!a-z]	Désigne un seul caractère n'appartenant pas à l'intervalle de caractères défini entre [ ]	
{frag1,frag2,..}	brase expansion	file_{one,two,three}

Tubes et les redirections

Utilisation des flux, des tubes (pipes) et des redirections

- **Description** : Les candidats doivent être capables de rediriger des flux et de les connecter dans le but de traiter efficacement ces données textuelles. Les tâches à effectuer comprennent les redirections de l'entrée standard, de la sortie standard et de la sortie standard des erreurs, connecter la sortie d'une commande à l'entrée d'une autre, utiliser la sortie d'une commande comme paramètre pour une autre commande et envoyer le résultat en même temps sur la sortie standard et dans un fichier.

- **Termes, fichiers et utilitaires utilisés :**
  - tee
  - xargs

Les tubes

- **Les tubes** Unix permettent de combiner des commandes en les utilisant comme des briques indépendamment de leur provenance.
- Trois types d'entrées / sorties
  - Entrée standard (**stdin**) : Descripteur **0**
  - Sortie standard (**stdout**) : Descripteur **1**
  - Sortie d'erreur standard (**stderr**) : Descripteur **2**

Redirection

Fonction de redirection	Syntaxe
Envoyer stdout vers file	$ cmd > file ou $ cmd 1> file
Envoyer stderr vers file	$ cmd 2> file
Envoyer stdout et stderr vers file	$ cmd > file 2>&1
Envoyer stdout vers file1 et stderr vers file 2	$ cmd > file1 2>file2
Recevoir stdin à partie de file	$cmd < file
Ajouter stdout à la fin du file	$ cmd >> file ou $ cmd 1>> file
Envoyer stderr à la fin du file	$ cmd 2>> file
Envoyer stdout et stderr à la fin du file	$ cmd >> file 2>&1

Les tubes (pipe)

- La sortie d'une commande devient l'entrée d'une autre.
- Tubes et redirections peuvent être combinées sur une ligne de commande selon les résultats qu'on veut obtenir.
- Pipe stdout de cmd1 vers cmd2 :

  $ cmd1 | cmd2

- Pipe stdout et stderr de cmd1 vers cmd2.

  $ cmd1 2>&1 | cmd2

- Afficher les 6 premières lignes du fichier /etc/passwd une fois ce fichier trié par ordre alphabétique

  $ sort /etc/passwd | head -6

- La commande tee permet de dupliquer le flux de données en sortie : **$ sort /etc/passwd | tee res1 | head -6**

La commande xargs

– **cmd1 | xargs cmd2**

- Permet de passer en arguments de la commande cmd2, le résultat de la commande cmd1

Exemples :

$ find /tmp -name core -type f -print | xargs /bin/rm -f

$ find /etc/ -name *.conf | xargs grep 'Linux'

Les Processus

Création, surveillance et destruction de processus

- **Description** :Les candidats doivent être capables d'effectuer une gestion de base sur les processus.

- **Termes, fichiers et utilitaires utilisés :**
  - &
  - bg
  - fg
  - jobs
  - kill
  - nohup
  - ps
  - top
  - free
  - uptime
  - killall

Processus Unix ?

- Un programme en cours d'exécution qui utilise les ressources de la mémoire + processeur.
- Quelques informations relatives à un processus :
  - **PID** : Process ID
  - **PPID** : Parent Process ID
  - User ID (**UID**) et Group ID (**GID**) : Ayant lancé le processus
  - temps CPU
  - tables de référence des fichiers ouverts

ps

- Quels sont les processus exécutés par le système
- Afficher tous les processus du système :

  # ps -A ou ps -ef
- **Manipulations**
  - l'utilisateur salah exécute la commande : **$ vi test**
  - Afficher les processus de l'utilisateur salah

    # ps -U salah
  - Afficher les utilisateurs qui exécutent la commande vi

    # ps -f -C vi

```
UID        PID  PPID  C STIME TTY          TIME CMD
salah     5229  5201  0 18:23 pts/5    00:00:00 vi tets
zied      5278  4370  0 18:31 pts/0    00:00:00 vi test
```

top

- Afficher des informations sur l'activité du système en temps réel

```
  3:37pm  up 46 days,  5:11,  2 users,  load average: 0.01, 0.17, 0.19
96 processes: 94 sleeping, 1 running, 0 zombie, 1 stopped
CPU states:  0.1% user,  1.0% system,  0.0% nice,  0.9% idle
Mem:  1030268K av,  933956K used,   96312K free,       0K shrd,  119428K buff
Swap: 1052216K av,    1176K used, 1051040K free                  355156K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
22069 ian       17   0  1104 1104   848 R     0.9  0.1   0:00 top
    1 root       8   0   500  480   444 S     0.0  0.0   0:04 init
    2 root       9   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root       9   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root      19  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root       9   0     0    0     0 SW    0.0  0.0   0:00 kswapd
```

- Quelques options interactives :
  - **ctrl-L** : refresh
  - **h** : help
  - **n** : nombre de processus à afficher
  - **q** : quitter
  - **r** : (renice) changer le priorité d'un processus

Envoyer un signal à un processus

- **kill [numéro-du-signal] PID**
- Afficher une liste des noms de signaux connu : **kill -l**

Signal name[a]	Number	Meaning and use
HUP	1	Hang up. This signal is sent automatically when you log out or disconnect a modem. It is also used by many daemons to cause the configuration file to be reread.
INT	2	Interrupt; stop running. This signal is sent when you type Ctrl-C.
KILL	9	Kill; stop unconditionally and immediately. Sending this signal is a drastic measure, as it cannot be ignored by the process. This is the "emergency kill" signal.
TERM	15	Terminate, nicely if possible. This signal is used to ask a process to exit gracefully.
TSTP	20	Stop executing, ready to continue. This signal is sent when you type Ctrl-Z.
CONT	18	Continue execution. This signal is sent to start a process stopped by SIGTSTP or SIGSTOP. (The shell sends this signal when you use the fg or bg commands after stopping a process with Ctrl-Z.)

Envoyer un signal à un processus (suite ...)

- Envoyer SIGTERM aux processus (PIDs 1000 et 10001

  $ kill 1000 1001

  $ kill -15 1000 1001

  $ kill -SIGTERM 1000 1001

  $ kill -TERM 1000 1001

- relecture des fichier de configurations

  kill -HUP `cat /var/run/httpd.pid`

- Arrêt forcé !

  kill -9 1000 1001 ou bien kill -KILL 1000 1001

- Afficher les processus qui s'exécutent en arrière plan (bg)

  # firefox &

  [1] 5788

  # jobs

  [1]+ Running ./firefox &

Envoyer un signal à un processus (encore ...)

- Vous avez oublié de lancer firefox en arrière plan (bg) :

```
# firefox
(vous faites ctrt z)
[1]+  Stopped                 firefox
# bg
[1]+ firefox &
```

TSTP (20)

CONT (18)

- Envoyer un signal à des processus indiqués par leurs noms
  - **killall -HUP httpd**

Modifier la priorité d'un processus

Modification des priorités des processus

- **Description** : Les candidats doivent être capables de gérer les priorités des processus.
- **Termes, fichiers et utilitaires utilisés :**
  - nice
  - ps
  - renice
  - top

Priorité des processus

- **top** ou bien **ps -l**
- le noyau offre **+ temps CPU** pour « hight priority process »
- Par défaut les processus d'un utilisateur sont créés avec **la Nice Number 0**.
- **Nice Number positif --> moins de priorité**
- **Nice Number négatif --> plus de priorité**
- **Nice Number** varie de **-20 à 19**
- Un utilisateur peut lancer un processus avec un **Nice Number** positif
- **SEUL root** peut lancer un processus avec un **Nice Number** négatif

nice et renice

- Un utilisateur lance cmd avec un **Nice Number** +5

  $nice -5 cmd1

- **Seul root** peut lancer des processus avecun **Nice Number** négatif

  # nice - -10 vi /etc/hosts.deny

  # nice -n -10 vi /etc/hosts.deny

- **renice : Modifier la priorité d'un processus**

  #renice -20 501

  #renice -10 -u salah -p 501

Recherche sur des fichiers texte avec des expressions régulières

Recherche dans des fichiers texte avec les expressions régulières

- **Description** : Les candidats doivent être capables de gérer les priorités des processus.
- **Termes, fichiers et utilitaires utilisés :**
  - grep
  - egrep
  - fgrep
  - sed

Objectifs et outils

- Recherche (texte) sur le contenu des fichiers, selon un modèle (motif) : « les expressions régulières » **regex**
- Une expression régulières (regular expression) est un motif qui permet de décrire un ensemble de chaînes.
- Outils : grep, egrep, sed, awk, Perl, java ...
- **grep [options] regex [fichiers]**
- **options :**

  -c : Afficher le nombre de lignes qui satisfait regex, pas les lignes

  -h :Ne pas afficher le nom des fichiers dans les résultats lorsque plusieurs fichiers sont parcourus.

  -i : Ignorer les différences majuscules/minuscules dans la recherche.

  -n : Ajouter à chaque ligne de sortie un préfixe contenant son numéro dans le fichier

  -v : Afficher les lignes qui **ne satisfait pas** regex

  -E : Interpréter regex comme une expression régulière étendu. **egrep**

Expression régulière : position

Regular expression	Description
^	Match at the beginning of a line. This interpretation makes sense only when the ^ character is at the left-hand side of the *regex*.
$	Match at the end of a line. This interpretation makes sense only when the $ character is at the right-hand side of the *regex*.
\<\>	Match word boundaries. Word boundaries are defined as whitespace, the start of line, the end of line, or punctuation marks. The backslashes are required and enable this interpretation of < and >.

Expression régulière : groupe de caractères

Regular expression	Description
`[abc][a-z]`	Single-character groups and ranges. In the first form, match any single character from among the enclosed characters `a`, `b`, or `c`. In the second form, match any single character from among the range of characters bounded by `a` and `z` (POSIX character classes can also be used, so `[a-z]` can be replaced with `[[:lower:]]`). The brackets are for grouping only and are not matched themselves.
`[^abc][^a-z]`	Inverse match. Match any single character not among the enclosed characters `a`, `b`, and `c` or in the range `a-z`. Be careful not to confuse this inversion with the anchor character `^`, described earlier.
`.`	Match any single character except a newline.

Expression régulière : Les modificateurs

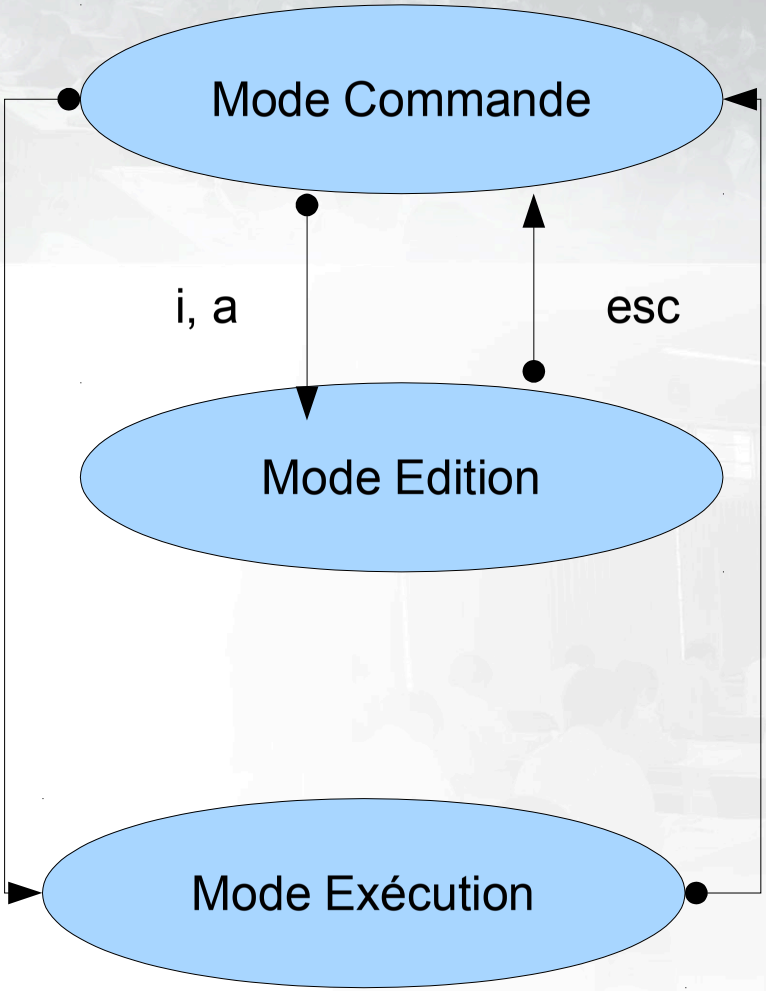
Basic regular expression	Extended regular expression (egrep)	Description
*	*	Match an unknown number (zero or more) of the single character (or single-character *regex*) that precedes it.
\?	?	Match zero or one instance of the preceding *regex*.
\+	+	Match one or more instances of the preceding *regex*.
\{*n*,*m*\}	{*n*,*m*}	Match a range of occurrences of the single character or *regex* that precedes this construct. \{*n*\} matches *n* occurrences, \{*n*,\} matches at least *n* occurrences, and \{*n*,*m*\} matches any number of occurrences from *n* to *m*, inclusively.
|	|	Alternation. Match either the *regex* specified before or after the vertical bar.
\(*regex*\)	(*regex*)	Grouping. Matches *regex*, but it can be modified as a whole and used in back-references. (\1 expands to the contents of the first \(\) and so on up to \9.)

L'éditeur vi

Édition de fichiers texte avec "vi"

- **Description** : Les candidats doivent être capables de gérer les priorités des processus.
- **Termes, fichiers et utilitaires utilisés :**
  - vi
  - /, ?
  - h,j,k,l
  - i, o, a
  - c, d, p, y, dd, yy
  - ZZ, :w!, :q!, :e!

L'éditeur historique du système UNIX



- x : supprimer caractère
- dd : supprimer la ligne
- u : undo
- ctrl-R : redo
- / : recherche
- n : résultat suivant

:

- w :enregistrer
- wq : enregistrer et quitter
-