

Évaluation

Durée : 50 minutes

Répondez directement dans ce fichier en complétant les blancs (signalés par un surlignage jaune).

Envoyez ce document complété en pièce jointe à aline.hufschmitt@iedparis8.net en précisant « REGROUPEMENT IED SEPTEMBRE 2017 » comme sujet du mail.

Nom	Prénom	Numéro étudiant

Culture générale (1 points)

Que signifie l'acronyme LUG (en anglais et en français) ?

Linux User Group : Groupe d'Utilisateur de Linux

Que signifie le sigle LTS (en anglais et en français) ?

Long Time Support : support sur le long terme

Qualifie certaines version de Linux qui proposent des mises à jour et un suivi de 5 ans

Qu'est-ce que le "lorem ipsum" ?

C'est un texte latin (de Cicéron) utilisé comme texte de remplissage afin de tester une mise en page

Quelle est la différence entre un éditeur de texte et un traitement de texte ?

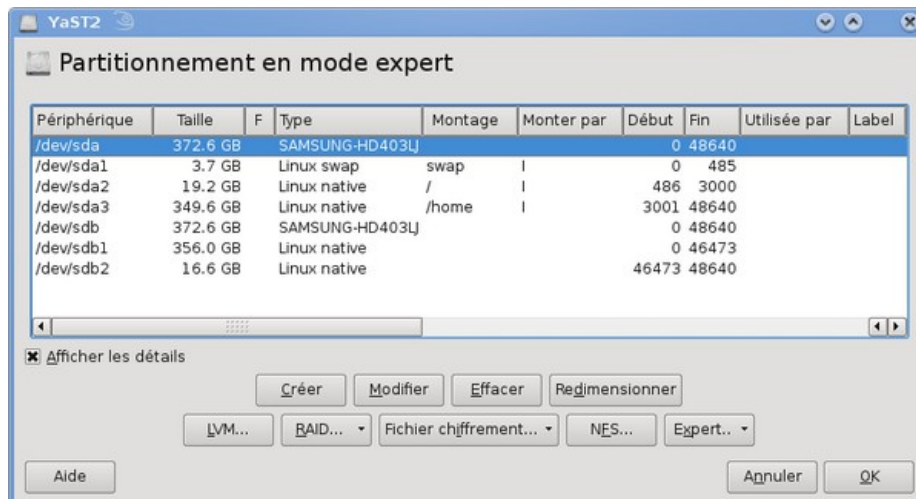
Un éditeur de texte permet d'éditer un texte sans aucune mise en page alors qu'un traitement de texte permet de spécifier différentes typographies de mettre en forme le texte dans les listes à points, tableaux, etc. Un traitement de texte enregistre le document créé dans un format qui permet de préciser son contenu et sa mise en page, contrairement à l'éditeur de texte qui enregistre le contenu au format « texte pur ».

Dans un document Latex, j'ai placé des ancrs avec la balise `\label{}` et j'y fait référence avec `\ref{}` plus loin dans le texte. Après compilation de mon document, toutes les références sont remplacées par des "??". Quelle erreur ai-je commise ?

Vous avez oublié de compiler 2 fois le fichier latex pour produire un fichier .aux contenant les références puis l'utiliser afin de remplacer les ?? par les informations adéquates.

A quoi sert le fichier .bashrc (en trois mots) et où s'attend on à le trouver ?

Ce fichier contient différentes commandes pour la configuration du Shell Bash, il permet de personnaliser son Shell et il est situé dans le home de chaque utilisateur (~)



D'après l'image ci-dessus, sur cette machine, combien y a-t-il

- de disques ? 2
- de partitions par disque ? 3 sur le premier 2 sur le second
- de quel type sont ces partitions ? Surement des partitions primaires

Utiliser le manuel (2 points)

Il existe plusieurs fonctions de la librairie permettant de manipuler les expressions régulières (regex). Comment trouver les pages de documentation sur ces fonctions de la librairie (et seulement elles) à partir du mot clef « regex » ? indiquez la commande et le résultat obtenu.

\$ man -s 3 -k regex

```
re_comp (3)      - BSD regex functions
re_exec (3)      - BSD regex functions
regcomp (3)      - POSIX regex functions
regerror (3)     - POSIX regex functions
regex (3)        - POSIX regex functions
regexexec (3)    - POSIX regex functions
regfree (3)      - POSIX regex functions
```

Quelle commande saisir pour trouver des informations sur la commande "locate" dans le manuel ?

\$ man locate

D'après la manuel à quoi sert cette commande ?

A rechercher des noms de fichiers correspondant à une recherche par mot clef (avec meta-caractères) ou bien avec une regex.

La commande a besoin de lire des informations dans une base de donnée, d'après le manuel, comment créer cette base ?

En utilisant la commande « updatedb »

Les licences (1 point)

Pour chacune de ces licences, indiquez brièvement pour quel usage elle a été conçue principalement (ce qu'elle protège ou ses particularités si elle en a).

Licence	Que protège t'elle ?	Est-elle copyleft ?
GNU-GFDL	Des documentations, tutoriels, etc.	Oui / Non
CeCILL	Des logiciels. Elle est prévue spécifiquement pour être en accord avec le droit français.	Oui / Non
BSD	Des logiciels.	Oui / Non

Que signifie copyleft ?

Que si on réutilise tout morceau ou l'intégralité du logiciel libre protégé dans un autre logiciel, celui ci sera également un logiciel libre protégé par la même licence : on ne peut pas le rendre propriétaire.

Quelle est, grossièrement (le sujet est vaste, inutile de trop détailler), la différence entre « libre » et « open source » ?

Open source signifie que le code est visible, qu'on peut l'étudier et rien de plus. Libre est quasiment une « philosophie », dans un logiciel libre non seulement le code est visible et peut être étudié mais il peut aussi être utilisé comme on le souhaite, modifié et redistribué.

Processus (1 points)

Complétez les blancs en indiquant les raccourcis claviers ou les commandes appropriées :

Pour stopper proprement l'interpréteur Python ou Lisp : **Ctrl + D**

Pour stopper en force un processus qui tourne en boucle : **Ctrl + C**

Pour mettre un processus en pause à l'arrière plan : **Ctrl + Z**

Pour examiner les ressources consommées par les processus : **\$ top (ou atop, htop)**

Pour récupérer le PID d'un processus tournant en arrière plan dans la console :

\$ **ps -a**

Pour arrêter en force un processus planté (PID=23452) :

```
$ kill -9 23452
```

Se déplacer, manipuler les fichiers (5 points)

Indiquez les commandes à saisir :

Pour se déplacer dans son répertoire personnel : `$ cd ~`

Pour revenir dans le répertoire parent : `$ cd ..`

Pour créer les fichiers `tic`, `tac` et `toe` : `$ touch tic tac toe`

Pour renommer le fichier `tic` en `le_tic` : `$ mv tic "le_tic"`

Pour supprimer le fichier `tac` : `$ rm tac`

Pour créer un dossier nommé `test` : `$ mkdir test`

Pour créer un chemin `test/script/shell` : `$ mkdir -p test/script/shell`

Pour savoir où on est : `$ pwd`

Pour lister les fichiers du répertoire courant en affichant les fichiers cachés : `$ ls -a`

Pour effacer le répertoire `script/shell` qui contient des fichiers : `$ rm -r script/shell`

Pour supprimer le répertoire vide `test` : `$ rmdir test`

Pour visualiser le contenu du fichier texte `~/ .bashrc` : `$ cat ~/ .bashrc`

Pour connaître le type du fichier `bin/date` : `$ file bin/date`

Pour comparer deux fichiers `tic` et `toe` : `$ diff tic toe`

Si je suis dans le répertoire `/home/damien/documents` et que je veux aller dans `/home/arthur/images`, indiquez la commande à saisir pour m'y rendre

- avec un chemin absolu : `cd /home/arthur/images`

- avec un chemin relatif : `cd ../../arthur/images`

Liens, redirections, pipes (2 points)

Indiquez les commandes à saisir :

Pour créer un raccourci nommé `doc` vers `/home/arthur/Document` :

```
$ ln -s /home/arthur/Document doc
```

Pour afficher le message `bonjour` et le rediriger dans un fichier nommé `message.txt` :

```
$ echo bonjour > message.txt
```

Pour mettre à jour la liste des paquets disponibles dans les dépôts avec `apt` et rediriger la sortie d'erreur vers un fichier `log.txt` :

```
$ apt-get update >2 log.txt
```

Pour exécuter le script `script.sh` situé dans le répertoire courant, rediriger la sortie vers un fichier `result.txt` et les erreurs au même endroit :

```
$ ./script.sh > result.txt 2>&1
```

Pour faire la même chose, mais sans écraser le contenu du fichier :

```
$ ./script.sh >> result.txt 2>&1
```

Pour trier le fichier `listing.txt`, supprimer les lignes en double, et afficher les 10 dernières lignes (en une seule commande) :

```
$ sort listing.txt | uniq | tail -n 10
```

Permissions, propriété, chemins d'accès (4 points)

Indiquez les commandes à saisir :

Pour donner la propriété du fichier `secret.sh` à l'administrateur et à son groupe habituel :

```
$ chown root:root secret.sh
```

Pour voir les droits des fichiers du répertoire courant :

```
$ ls -l
```

D'après la ligne suivante :

```
drwxr-xr-- 6 arthur staff 204 26 jui 10:11 WWW
```

Quel est le type de fichier ? **c'est un répertoire**

Qui a le droit de modifier ce fichier ? **Arthur uniquement (et l'admin)**

Comment créer un lien physique vers ce fichier ? **On ne peut pas**

Qui n'est pas autorisé à voir le contenu de ce fichier ? **Il faut le droit « x » pour voir le contenu d'un répertoire, donc les utilisateurs qui ne sont pas arthur et pas dans le groupe staff ne peuvent pas voir le contenu**

Indiquez la commande à saisir pour interdire à Arthur de modifier ce fichier, sans modifier le reste :

```
chmod u-w WWW
```

Pour savoir quel Shell utilise Arthur, dans quel fichier puis-je aller voir ?

```
Dans /etc/passwd
```

Si je crée ma propre commande `date` placée dans le répertoire courant, comment lancer cette commande pour être sûr que c'est elle que j'exécute et non celle du système ?

```
$ ./date
```

Quelle est la base de l'arborescence d'un système de fichier GNU/Linux ? **/**

A quoi sert la variable `PATH` de l'environnement ? **Elle contient les chemins vers les répertoires contenant les commandes et applications. Elle permet de trouver une commande à exécuter juste à partir de son nom même si on ne précise pas où elle se trouve.**

Expressions régulières (2 point)

Voici un listing contenu dans le fichier `listing.bas` :

```
10 INPUT "Quel est votre nom ? "; NOM$
20 PRINT "Bonjour "; NOM$
30 INPUT "Combien d'étoiles voulez-vous ? "; NOMBRE
40 LET ETOILE$ = ""
50 FOR I = 1 TO NOMBRE
60 LET ETOILE$ = ETOILE$ + "*"
70 NEXT I
80 PRINT ETOILE$
90 INPUT "Voulez-vous plus d'étoiles ? "; ETOILE$
100 IF LEN(ETOILE$) = 0 THEN GOTO 90
110 LET ETOILE$ = LEFT$(ETOILE$, 1)
120 IF (ETOILE$ = "0") OR (ETOILE$ = "o") THEN GOTO 30
130 PRINT "Au revoir ";
140 FOR I = 1 TO 200
150 PRINT NOM$; " ";
160 NEXT I
170 LPRINT
```

Ecrivez une commande avec `grep` pour récupérer les lignes commençant par les nombres de 50 à 130 inclus :

```
$ grep -E "^( [5-9]0|1[1-3]0)" listing.bas
```

Ecrivez une commande avec `grep` pour récupérer toutes les lignes contenant un point d'interrogation :

```
$ grep -E "?" listing.bas
```

Ecrivez une commande avec `grep` pour récupérer toutes les lignes contenant le mot `PRINT` (mais pas `LPRINT`) :

```
$ grep -E " PRINT" listing.bas ou grep -E "\<PRINT" listing.bas
```

Ecrivez une commande avec `grep` pour récupérer toutes les lignes s'achevant par un nombre :

```
$ grep -E "[0-9]+$"
```

Script Shell (2 points)

Ecrire une commande Shell qui compte à rebours de 10 à 0 en attendant une seconde (environ) entre chaque nombre affiché (indice : commande `sleep`) :

```
$ for i in {10..0} ; do sleep 1 ; echo $i ; done
```

Écrire un script Shell qui simule un jeu de Quizz avec une seule question. Voici un exemple du comportement attendu pour votre script (en bleu les saisies de l'utilisateur) :

```
$ ./test.sh
L'ASM est un langage assembleur (vrai/faux) ? vrai
bravo !

$ ./test.sh
L'ASM est un langage assembleur (vrai/faux) ? faux
eh non !
```

```
#!/bin/sh

read -p "L'ASM est un langage assembleur (vrai/faux) ? " REP

if [ $REP = "vrai" ] ; then
    echo "bravo !"
else
    echo "eh non !"
fi
```