

## CONTRÔLE TERMINAL -L2 INFO - SYSTÈMES D'EXPLOITATION

Durée de l'épreuve = 2h, Documents interdits sauf « mini tutoriel bash »

### Exercice 1 (7 points)

1. Écrire un script nommé `ex01` qui crée le répertoire `ZIPTXT` dans votre dossier personnel (`$HOME`). Si le répertoire existe déjà, vous devrez afficher un message d'erreur et le script doit s'arrêter.
2. Le script doit ensuite prendre tous les fichiers d'extension `.txt` du répertoire `$HOME` puis les zipper et les déplacer dans le répertoire `$HOME/ZIPTXT`
3. Le script doit enfin créer un fichier `$HOME/ZIPINFO.txt` qui doit contenir la liste de tous les fichiers zippés à la question précédente

### Exercice 2 (6 points)

Écrire un script shell bash nommé `ex02` qui :

- prend un nombre `n` comme unique paramètre. S'il y a plus ou moins d'un paramètre, vous devrez afficher le message « trop ou pas assez de paramètres ».
- crée `n` fichiers nommés `fic1.txt`, `fic2.txt`, ... `ficn.txt` en y mettant la chaîne de caractères « `bonjour` » puis les ouvre avec le logiciel `gedit`.

### Exercice 3 (7 points)

Écrire un script shell bash nommé `ex03` qui :

- prend un nom d'utilisateur comme unique paramètre. (on ne vérifiera pas qu'il n'y a qu'un paramètre et qu'il correspond à un utilisateur )
- affiche l'id , le shell utilisé par cet utilisateur ainsi que les groupes auxquels il appartient.

Pour obtenir ces informations, votre programme ira **obligatoirement** lire :

- le fichier `/etc/passwd` dont les 3ème et dernier champs sont l'id et le shell utilisé (vous ne pouvez pas utiliser la commande `id`)
- le fichier `/etc/group` dont le premier champ contient le nom du groupe et le dernier champ contient les utilisateurs qui y sont inscrits séparés par une virgule (vous ne pouvez pas utiliser la commande `groups`)

**Exemple :** la commande `ex03 toto` affiche « `toto a pour id 1001 , a pour shell /bin/bash et appartient aux groupes : toto admin` »

On veillera à ce que l'affichage du résultat ne contienne pas de retours à la ligne entre le nom des groupes.

## FORMULAIRE COMMANDES DE BASE UNIX

<b>Manipulation de fichiers</b> <b>cat fic</b> : affiche le contenu du fichier <b>fic</b> . <b>more fic</b> : idem mais fait une pause entre chaque page. <b>mv fic rep</b> : déplace le fichier <b>fic</b> dans le répertoire <b>rep</b> . <b>mv fic1 fic2</b> : renomme le <b>fic1</b> en l'appelant <b>fic2</b> . <b>cp fic1 fic2</b> : fait une copie de <b>fic1</b> en l'appelant <b>fic2</b> . <b>cp fic rep</b> : fait une copie de <b>fic</b> et la place dans le dossier <b>rep</b> . <b>rm fic</b> : Supprime le fichier <b>fic</b> . <b>touch fic</b> : créé un fichier vide nommé <b>fic</b> <b>sort fic</b> : trie le contenu du fichier <b>fic</b> <b>uniq fic</b> : supprime les doublons consécutifs du fichier <b>fic</b> <b>cut -f2 -d: fic</b> : filtre le champ 2 du fichier <b>fic</b> avec le délimiteur : <b>wc fic</b> : compte les mots, caractères et lignes du fichier <b>fic</b> . <b>grep coucou fic</b> : affiche les lignes du fichier <b>fic</b> qui contiennent le mot COUCOU <b>head fic</b> : affiche le début du fichier <b>fic</b> . <b>tail fic</b> : affiche la fin du fichier <b>fic</b> .	<b>Manipulation de dossiers (répertoires)</b> . désigne le répertoire courant <b>pwd</b> : affiche le chemin absolu du répertoire courant. <b>ls</b> : affiche le contenu du répertoire courant. <b>ls rep</b> : affiche le contenu du répertoire <b>rep</b> . <b>ls -al rep</b> : affiche le contenu du répertoire <b>rep</b> avec plus de détails (l) et avec les fichiers cachés (a). <b>cd</b> : Vous place dans votre répertoire personnel. <b>cd rep</b> : vous place dans le répertoire <b>rep</b> . <b>cd -</b> : Vous place dans le répertoire où vous étiez précédemment. <b>cd ~</b> : vous place dans votre répertoire personnel <b>cd ..</b> : vous place dans le répertoire parent du répertoire courant <b>mkdir rep</b> : créé le répertoire <b>rep</b> . <b>rmdir rep</b> : efface le répertoire <b>rep</b> s'il est vide. <b>rm -r rep</b> : efface le répertoire <b>rep</b> et tout son contenu.  `` <b>(backquote)</b> : pour lancer des commandes à insérer dans une chaîne de caractères : Exemple : <code>echo mon nom est `whoami`</code>
<b>Redirection d'entrées-sorties</b> <b>com1 &gt; fic</b> : écrit le résultat de la commande <b>com1</b> dans le fichier <b>fic</b> . <b>com1 &gt;&gt; fic</b> : écrit le résultat de la commande <b>com1</b> à la fin du fichier <b>fic</b> . (sans l'écraser) <b>com1   com2</b> : envoie le résultat de la commande <b>com1</b> en entrée de la commande <b>com2</b> .	<b>Jokers</b> Le caractère <b>*</b> remplace n'importe quelle suite de lettres Le caractère <b>?</b> remplace n'importe quelle lettre mais une seule. <b>[a-f]</b> désigne les caractères compris entre a et f