

IMA 4 - T.P. Système - #1

1 Exploitation et combinaison de filtres

Donnez les commandes `Unix` qui permettent de répondre aux objectifs suivants :

1. visualiser le répertoire `/usr/bin` page par page
2. stocker l'aide de la commande `cat` dans le fichier `catAide.txt`
3. à l'aide de la commande `cat` construire le fichier texte de nom `titreCat.txt` contenant le texte suivant :

```
-----  
- COMMANDE CAT -  
- EXPLICATIONS -  
-----
```

4. écrire dans le fichier `catTitreAide.txt` la concaténation des fichiers `titreCat.txt` et `catAide.txt`
5. visualisez le fichier `strange.txt`. Comme ce dernier n'existe pas, que se passe-t-il ? Relancez cette commande en redirigeant la sortie d'erreur vers `/dev/null`, que se passe-t-il ? Relancez cette commande en redirigeant la sortie d'erreur vers le fichier `err.txt`
6. avec la commande `cat`, écrire le fichier suivant :

```
#include <stdio.h>  
int main()  
{  
    fprintf(stdout, "message sur la sortie standard\n");  
    fprintf(stderr, "message sur la sortie d'erreur\n");  
    return 0;  
}
```

puis le compiler et l'exécuter. Lors de l'exécution, voyez-vous une différence dans l'affichage ? pourquoi ? Redirigez maintenant la sortie standard vers `/dev/null`. Que constatez-vous ?

7. à partir du texte correspondant à l'aide de la commande `sort`, affichez uniquement les 5 dernières lignes contenant le mot "sort"
8. affichez, page par page, par ordre inverse lexicographique sur les noms de fichiers, tous les fichiers commençant par la lettre 'f' du répertoire `/dev` ainsi que les informations sur ces fichiers (droits, taille, ...)

2 Expressions rationnelles, grep, sort, sed, ...

2.1 Manipulation du fichier des pages imprimées

Lorsque vous imprimez un document sur l'une des imprimantes disponibles dans les salles de TP de Polytech, le nombre de pages imprimées vous est décompté. D'une façon plus générale toutes les pages imprimées sont mémorisées dans un fichier "log" de nom `/var/log/printaccounting` qui a l'allure suivante :

Thu Apr 6 10:30:23 CEST 2000|Vincent.Vandamme|hainaut.priv.eudil.fr|gayant|2

soit : date et heure|prénom.nom|nom_machine|nom_imprimante|nb_pages

le **nb_pages** reçoit le nombre de pages effectivement imprimées ou **aborted** ou **none** selon les cas d'échec à l'impression. Ce fichier `/var/log/printaccounting` se trouve sur le serveur hainaut et ne vous est donc pas accessible. Une copie partielle se trouve dans le répertoire `/usr/local/tmp/IMA`. Ne le recopiez pas chez vous car ce fichier est trop "gros". On se propose de répondre à quelques questions à propos de ce fichier de log :

1. Donnez le nombre de lignes de ce fichier
2. Donnez le nombre de fois où l'impression est "aborted"
3. Donnez le nombre de fois où l'impression est ni "aborted" ni "none"
4. Donnez le nombre d'impression "aborted" sur l'imprimante "reuze"
5. Donnez toutes les impressions à partir d'une machine kipper. Vérifiez bien, sur le résultat de ne pas avoir affiché les impressions de l'imprimante kipper
6. Vous constaterez que les machines sont nommées kipperXX et d'autres kipperXX.priv.eudil.fr. Donnez toutes les impressions qui concernent les machines kipperXX.
7. Donnez le nombre d'impression sur l'imprimante "gayant"
8. Donnez le nombre de pages imprimées sur l'imprimante "gayant"
9. Donnez le nombre total d'impression sur "gayant" et "reuze"
10. Donnez les 10 plus grosses impressions. Attention !! Cela peut prendre du temps !!
11. Listez les nombres de pages imprimées dans l'ordre décroissant. En cas d'égalité utilisez l'ordre alphabétique des noms d'imprimantes puis des **prenom.nom**. Réalisez un affichage page par page.
12. Transformez toutes les lignes "aborted" selon l'exemple suivant :
Thu Feb 3 11:54:31 CET 2000|Ahmed.Laib|gayant04|gayant|aborted devient Thu Feb 3 11:54:31
CET 2000|gayant|erreur

2.2 Comptes utilisateurs

La gestion des login et groupes des utilisateurs sont gérés par l'intermédiaire d'un système de base de données de type **Berkeley DB**. Une copie de la base des utilisateurs se trouve dans le fichier `/usr/nsswitch/temp_pat` et une copie des groupes se trouve dans le fichier `/usr/nsswitch/temp_groups`.

1. Pour comprendre les informations se trouvant dans les deux fichiers, affichez toutes les lignes comportants votre login. Pour visualiser le contenu des fichiers, vous devez utiliser la commande **db_dump -p**. Les lignes redondantes doivent être effacées.
2. À partir du fichier **temp_groups**, vous devez générer le fichier `/tmp/groups` contenant sur chaque ligne, le nom du groupe utilisateur suivi de son numéro. Les deux éléments sont séparés par le caractère ":". Par exemple, pour le groupe ima4, le résultat sera **ima4:1024**. Le fichier résultat sera trié par ordre des numéros de groupe.
3. À partir du fichier **temp_pat**, vous devez générer le fichier `/tmp/users` contenant sur chaque ligne les informations associés à un compte utilisateur. Le fichier sera trié par ordre des numéros de groupe.
4. Vous devez fusionner les fichiers `/tmp/groups` et `/tmp/users` afin de créer le fichier `/tmp/users_groups`
5. À partir du fichier `/tmp/users_groups` vous devez générer une liste des comptes utilisateurs selon le format suivant **Prenom Nom:Groupe**. Par exemple **Thomas Vantroys:imaEns**

3 Alias

1. Ouvrez un nouveau terminal et affichez son nom à l'aide de la commande `tty`. Vous allez définir un alias `dirr` permettant de lister tout le contenu de votre répertoire d'accueil (en format long et y compris les fichiers "systèmes") en y incluant tous les sous répertoires. Cet alias doit donner les mêmes résultats quelque soit le répertoire d'où vous exécutez `dirr`.
2. Ouvrez un nouveau terminal et affichez son nom à l'aide de la commande `tty`. Exécutez la commande `dirr` dans les deux terminaux. Que se passe-t-il ? Pourquoi ?
3. Listez les alias définis dans le premier terminal et dans le deuxième terminal. Dans le premier terminal, supprimez l'alias `dirr` et ajoutez dans votre fichier `.bashrc` la définition de l'alias `dirr`.
4. Ouvrez un troisième terminal. Testez la commande `dirr` dans les trois terminaux. Conclusion ?
5. Faites en sorte que la commande `dirr` soit désormais connue du deuxième terminal.

4 Les fichiers de commandes

Pour tous les fichiers de commandes à réaliser dans cette section, n'oubliez pas d'intégrer les conventions UNIX suivantes :

Un script shell subit les mêmes règles de programmation que n'importe quel langage (commentaires, indentation, déclaration de constantes, ...) et doit être lisible.

Tout programme UNIX retourne la valeur 0 lors d'une exécution correcte et une valeur distincte pour chaque erreur rencontrée.

1. À partir de votre répertoire d'accueil, créez un répertoire `bin`
2. Créez un script shell `analyseFichier.sh` qui affiche si le nom passé en paramètre est un fichier, un répertoire ou n'existe pas dans le répertoire courant (gérez tous les cas d'erreurs possibles : pas de paramètres, trop de paramètres, ...).
3. Rangez ce fichier de commande dans votre répertoire `bin` précédemment créé. Effectuez la commande `chmod u+x analyseFichier.sh`.
4. Faites en sorte que cette commande puisse être exécutée depuis n'importe quel répertoire sans devoir donner son chemin absolu ou relatif.
5. Créez le fichier de commande `analyseFichierRep.sh` qui affiche les mêmes renseignements que le fichier de commande précédent mais pour le fichier situé dans le répertoire passé en second paramètre. Ce fichier de commande **utilise** le script `analyseFichier.sh`.
6. Créez le script `analyseRep.sh` qui affiche les mêmes renseignements que précédemment mais pour tous les éléments du répertoire passé en paramètre de la commande.
7. Créez le script `analyseRepLu.sh` qui fait la saisie du nom d'un répertoire et appelle le script précédent.
8. Créez le script `analyseFichiers.sh` qui affiche les mêmes renseignements que précédemment mais pour tous les éléments passés en paramètre de la commande. Testez votre script :
`analyseFichiers.sh .*`
`analyseFichiers.sh *`
9. Créez le script `compteRep.sh` qui permet de connaître le nombre de fichiers et répertoires que contient un répertoire passé en paramètre de la commande.
10. Créez le script `analyseArbo.sh` qui affiche les renseignements (fichier ou répertoire) pour l'arborescence à partir du répertoire passé en paramètre.
11. Créez le script `compteArbo.sh` qui compte le nombre de fichiers et de répertoires pour l'arborescence à partir du répertoire passé en paramètre.
12. Créez le script `nettoyeur.sh` qui élimine les fichiers de nom `core`, les fichiers de suffixes `.bak`, `.*` pour l'arborescence à partir du répertoire passé en paramètre.
Important : pour éviter des pertes de fichiers irrémédiables, tester votre script avec la commande `echo` avant d'utiliser la commande `rm` !!!

5 Gestion de vos pages imprimées

Modifiez votre fichier de configuration `.profile` pour que lors de chaque lancement de votre shell, les informations suivantes s'affichent :

- Le nombre de pages que vous avez déjà imprimé
- le nombre de crédits que vous avez déjà utilisé
- le nombre de crédits qu'il vous restait après la dernière impression
- le nombre de crédits qu'il vous reste potentiellement. Pour simplifier, nous considérons que ce nombre augmente de 1 chaque jour

Pour cela, vous utiliserez le fichier `.printcredit.log` qui se trouve à la racine de votre compte. Dans ce fichier, chaque ligne correspond à une impression.

6 Les processus

Quand on lance une exécution en arrière plan, on obtient du système le numéro du processus. Ce numéro permet d'effectuer certaines opérations.

La commande `jobs` fournit la liste des travaux actifs ainsi que l'état de ces travaux. Le diagramme donné en cours décrit les commandes permettant de faire passer un processus d'un état à un autre.

Question préliminaire : écrire un fichier de commande, nommé `boucle.sh`, qui boucle sur une écriture écran (commande `echo`) et sur une mise en sommeil (commande `sleep`).

6.1 En mode interactif avec plusieurs terminaux disponibles

Lancez cette commande en mode interactif dans un terminal.

1. donnez une commande qui permet de limiter l'affichage aux processus qui vous appartiennent.
2. Donnez les relations de filiation qui lient votre processus au processus `init` ou `systemd` (en fonction de la version de votre système d'exploitation).
3. tuez le processus `boucle.sh` à l'aide de la commande `kill`. Affichez de nouveau la liste de vos processus.

6.2 En mode interactif avec un seul terminal disponible

Relancez le programme `boucle.sh` dans le même terminal.

1. comment suspendre le processus ainsi créé tout en restant dans le terminal ?
2. comment reprendre l'exécution du processus ainsi suspendu ?

6.3 En mode asynchrone (*batch*) avec plusieurs terminaux

Les différents terminaux que vous manipulez sont associés à des fichiers périphériques du répertoire `/dev`. Vous pouvez connaître le nom de votre terminal en utilisant la commande `tty`. Ainsi, le premier terminal sous X Window correspond à `/dev/pts/0`. Il est possible d'écrire dans ces fichiers comme on écrit dans n'importe quel fichier.

1. Relancez le programme `boucle.sh` en mode asynchrone à l'aide du symbole "&", dans un terminal.
2. le terminal est-il encore disponible pour lancer de nouvelles commandes ?

3. essayez de tuer ou suspendre le processus à l'aide des commandes `Ctrl` + `C` ou `Ctrl` + `Z`. Conclusion ?
4. Tuez le processus `boucle.sh` à l'aide de la commande `kill`
5. Comment lancer le programme `boucle.sh` en mode asynchrone dans un terminal tout en redirigeant la sortie standard vers un deuxième terminal ?
6. Créez un second fichier de commande `boucle2.sh` similaire au fichier de commande `boucle.sh` mais avec un affichage différent. Lancez ce programme, en mode asynchrone, dans le deuxième terminal en redirigeant la sortie vers le premier terminal.
7. Affichez la liste des travaux au moyen de la commande `jobs`. Relevez le numéro du job associé au processus `boucle.sh`. Suspendez l'exécution du processus `boucle.sh`. Affichez l'état des travaux.
8. Reprenez l'exécution de `boucle.sh` en arrière plan (asynchrone) avec la commande `bg`. Suspendez l'exécution du processus `boucle2.sh`. Continuez l'exécution. Continuez l'exécution de ce processus en avant-plan (interactif) avec la commande `fg`.
9. Tuez les 2 processus avec la commande `kill` appliquée au numéro de travail de chacun de ces processus.

7 Répertoire /proc

À chaque processus correspond un répertoire dans `/proc` ayant pour nom le PID du processus et contenant divers fichiers stockant les informations utilisées par le système afin de gérer le processus. Reportez-vous au `man` de `proc` pour plus d'informations.

1. Lancer le script `boucle.sh` en redirigeant la sortie standard vers `/dev/null` ;
2. Retrouvez dans le répertoire `/proc` les informations suivantes liées à l'exécution de `boucle.sh` :
 - le statut, le nombre de changements de contexte (cf fichier `status`) ;
 - le répertoire de travail, la commande de lancement du processus (cf `exe`, `cwd`)