

Sujet d'Examen du 20 juin 2007

Durée: 2 heures (+ 15 minutes pour la lecture du sujet)

Documents autorisés, mais non transmissibles

Sujet en 2 parties

Qualités appréciées :

- Lisibilité de la copie, concision, précision, exhaustivité.

Conseils :

- Que le manque de précision ne vous bloque pas malgré tout dans vos écrits.

N'hésitez pas à vous exprimer (même avec un doute, que vous pouvez noter par '?' dans la marge), surtout pour ce genre d'examen en temps limité.

- Par contre, pas de temps pour écrire des généralités ! Pas de temps pour être hors sujet !

- Inutile également de répéter le sujet, mais respecter et répéter la numérotation des questions.

Partie I	Gérard Nowak
Unix: Shell scripts et expressions régulières Système de fichiers	
(sujet: 6 pages)	(50 minutes - 8 points)

I.1 Shell scripts et expressions régulières (30 minutes - 5 points)
--

Question:

1) – (lire 'Annexe de I.1')

Soit un fichier non trié d'enregistrements, chacun contenant le nom, le numéro d'étudiant et le ou les prénoms (dans cet ordre) d'utilisateurs d'un groupe d'utilisateurs Unix.

Nom, numéro et prénom(s) sont séparés par le caractère '\$'.

Nom et prénom peuvent être composés de une ou plusieurs parties séparées par le caractère ' '.

Chaque partie de nom et prénom peut contenir les caractères '-' et ''.

Nom et prénom ne contiennent pas de lettres accentuées.

A l'aide des commandes Unix: cat, grep, tr, sort, diff, et des possibilités internes à Shell: while, read, les redirections de flux, la variable IFS (Internal Field Separator), echo, if, test, exit, écrire le Shell script de nom traiter_NP (une trentaine de lignes) qui reçoit le nom du fichier, et produit plusieurs fichiers.

1^{er} fichier) - celui de même nom mais suffixé par "_doublons" des enregistrements présents au moins de façon double, obtenu par production brute de diff (cf. option u de sort).

Après cette étape, nous travaillons pour la suite sur un fichier sans doublons.

2^e fichier) - celui de même nom mais suffixé par "_err" des enregistrements syntaxiquement incorrects.

Un enregistrement est syntaxiquement correct:

- s'il contient au moins 3 champs
(1 nom, 1 numéro d'exactly 6 chiffres décimaux, au moins 1 prénom)
- si chaque champ: nom, ou prénom, commence par une lettre majuscule ou minuscule et se poursuit éventuellement, avec lettre(s) ou tiret(s) ou apostrophe(s) ou espace(s).

Avec ces règles, on remarque et on accepte que, par exemple, a-B-c'd e---F est un nom syntaxiquement correct.

Après cette étape, nous travaillons pour la suite sur un fichier sans enregistrements erronés.

3^e fichier) - celui de même nom mais suffixé par "_probeOFF" (obtenu par diff) et qui doit être vide si la concaténation des fichiers d'enregistrements syntaxiquement erronés et corrects correspond au fichier sans doublons d'enregistrements. Si le fichier de preuve n'est pas vide, le Shell script se termine avec le code retour 3, et c'est l'indication qu'il n'est pas encore correct.

4^e) - celui de même nom mais suffixé par "_email_addresses" des enregistrements de type adresse électronique issus des enregistrements syntaxiquement corrects.

Une adresse électronique est formée ainsi:

premier_prénom.nom@domaine

avec domaine étant: li350.

prénom et nom contiennent des '-' à la place de '.' et '@'.

5^e fichier) - celui de même nom mais suffixé par "_out" des enregistrements syntaxiquement corrects triés par ordre alphabétique des noms et contenant dans cet ordre: numéro, séparateur '\$ ', nom, séparateur '\$ ', prénom (ou prénoms séparés par '\$ ').

A ce point, il reste malgré tout un problème potentiel d'homonymie.

Des enregistrements syntaxiquement corrects sont homonymes s'ils contiennent les mêmes noms et prénoms (selon leur ordre originel) bien qu'ayant des numéros de dossiers différents.

Le script produira aussi:

6^e fichier) - celui de même nom mais suffixé par "_homonymous" (obtenu par production brute de diff), des enregistrements homonymes.

Ce script rend:

- 0 si tout s'est correctement exécuté,
- 1 s'il manque l'argument nom du fichier,
- 2 si le fichier à trier n'existe pas,
- 3 si la séparation en enregistrements syntaxiquement corrects et erronés est fautive.

Annexe de I.1 :

Extrait de 'man bash':

IFS The Internal Field Separator that is used for word splitting after expansion and to split lines into words with the read builtin command. The default value is "<space><tab><newline>".

Extraits de 'man sort':

NAME

sort - sort lines of text files

SYNOPSIS

sort [OPTION]... [FILE]...

DESCRIPTION

Write sorted concatenation of all FILE(s) to standard output.

Ordering options:

-f, --ignore-case

fold lower case to upper case characters

-k POS1[,POS2]

an alternate syntax for specifying sorting keys. The fields and character positions are numbered starting with 1

-o, --output=FILE

write result to FILE instead of standard output

-t separator

use character separator as the field separator when finding the sort keys in each line

-u for the default, only output the first of a sequence of lines that compare equal

Extraits de 'man grep':

NAME

grep - print lines matching a pattern

SYNOPSIS

grep [options] PATTERN [FILE...]

DESCRIPTION

Grep searches the named input FILEs (or standard input if no files are named, or the file name - is given) for lines containing a match to the given PATTERN. By default, grep prints the matching lines.

OPTIONS

-E, --extended-regexp Interpret PATTERN as an extended regular expression.

-v, --invert-match Invert the sense of matching, to select non-matching lines.

REGULAR EXPRESSIONS

A regular expression is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions.

A regular expression may be followed by one of several repetition operators:

? The preceding item is optional and matched at most once.

* The preceding item will be matched zero or more times.

+ The preceding item will be matched one or more times.

{n} The preceding item is matched exactly n times.

{n,} The preceding item is matched n or more times.

{n,m} The preceding item is matched at least n times, but not more than m times.

A whole subexpression may be enclosed in parentheses.

Extraits de 'man tr':

NAME

tr - translate characters

SYNOPSIS

tr [-Ccsu] string1 string2

DESCRIPTION

The tr utility copies the standard input to the standard output with substitution of selected characters.

The characters in string1 are translated into the characters in string2 where the first character in string1 is translated into the first character in string2 and so on.

Extraits de 'man diff':

NAME

diff - compare files line by line

SYNOPSIS

diff [OPTION]... FILES

FILES are `FILE1 FILE2' or `DIR1 DIR2' or `DIR FILE...' or `FILE... DIR'.

If a FILE is `-', read standard input.

Extraits de 'man cat':

NAME

cat - concatenate and print files

SYNOPSIS

cat [-benstuv] [-] [file ...]

DESCRIPTION

The cat utility reads files sequentially, writing them to the standard output. The file operands are processed in command line order. A single dash represents the standard input.

Exemple:

Fichier "NP"

```
Duchemin$234567$Alexandra
Abd'oun$123456$Hubert$Marc-Alain
Le Corre$099990$Jean Julien$Alix
de Begmeil du Crac'h$990087$Pierre-Yves$Jean-Daniel$Leon
Abd'oun$123a56$Hubert$Marc-Alain
de Begmeil du Crac'h$99008777$Pierre-Yves$Jean-Daniel$Leon
Abd'oun$123456$Hub666ert$Marc-Alain
Duchemin$234567$Alexandra
de Begmeil du Crac'h$990087XXX$Pierre-Yves$Jean-Daniel$Leon
Abd'oun$123456$Hubert$Marc-Alain
de Begmeil du Crac'h$990087$Pierre-Yves$Jean-Daniel$Leon
Duchemin$234569$Alexandra$zoe
Abd'oun$123456$
Duchemin$234567$Alexandra
Abd'oun$123456$Hub666ert$Marc-Alain
$990087$Pierre-Yves$Jean-Daniel$Leon
Duchemin$234467$Alexandra
de Begmeil du Crac'h$991187$Pierre-Yves$Jean-Daniel$Leon
Abd'oun$123456
de Begmeil du Crac'h$087$Pierre-Yves$Jean-Daniel$Leon
```

Fichier "NP_doublons" (production brute de 'diff')

```
5,6d4
< Abd'oun$123456$Hub666ert$Marc-Alain
< Abd'oun$123456$Hubert$Marc-Alain
11,12d8
< Duchemin$234567$Alexandra
< Duchemin$234567$Alexandra
17d12
< de Begmeil du Crac'h$990087$Pierre-Yves$Jean-Daniel$Leon
```

Fichier "NP_err"

```
$990087$Pierre-Yves$Jean-Daniel$Leon
```

Abd'oun\$123456
Abd'oun\$123456\$
Abd'oun\$123456\$Hub666ert\$Marc-Alain
Abd'oun\$123a56\$Hubert\$Marc-Alain
de Begmeil du Crac'h\$087\$Pierre-Yves\$Jean-Daniel\$Leon
de Begmeil du Crac'h\$99008777\$Pierre-Yves\$Jean-Daniel\$Leon
de Begmeil du Crac'h\$990087XXX\$Pierre-Yves\$Jean-Daniel\$Leon

Fichier "NP_email_addresses"

Hubert.Abd-oun@li350
Alexandra.Duchemin@li350
Alexandra.Duchemin@li350
Alexandra.Duchemin@li350
Jean-Julien.Le-Corre@li350
Pierre-Yves.de-Begmeil-du-Crac-h@li350
Pierre-Yves.de-Begmeil-du-Crac-h@li350

Fichier "NP_out"

123456 \$ Abd'oun \$ Hubert \$ Marc-Alain
990087 \$ de Begmeil du Crac'h \$ Pierre-Yves \$ Jean-Daniel\$Leon
991187 \$ de Begmeil du Crac'h \$ Pierre-Yves \$ Jean-Daniel\$Leon
234467 \$ Duchemin \$ Alexandra
234567 \$ Duchemin \$ Alexandra
234569 \$ Duchemin \$ Alexandra \$ zoe
099990 \$ Le Corre \$ Jean Julien \$ Alix

Fichier "NP_homonymous" (production brute de 'diff')

3d2
< Duchemin\$Alexandra
7d5
< de Begmeil du Crac'h\$Pierre-Yves\$Jean-Daniel\$Leon

Fichier "NP_probeOFF" (vide ici)

***** Fin de l'Annexe de I.1 *****

I.2 Système de fichiers (20 minutes - 3 points)

Soit un utilisateur ordinaire en session ordinaire, juste après l'authentification réussie.
Et soit la suite de commandes suivante (elles sont numérotées pour faciliter leur référence):

- (0) umask 027
- (1) mkdir 200607 ; cd 200607
- (2) echo > ./D0/f1
- (3) echo "Hello John" >> f2
- (4) mkdir D1 ; cat f2 f2 >> D1/f3
- (5) ln -s ../D0/f1 sln1
- (6) mv f2 .f22
- (7) ln -s D1/f3 sln2
- (8) ln .f22 D1/hln1
- (9) ln D1 D1/hln2

Ces commandes sont exécutées dans l'ordre proposé.

Questions:

1) – Dessiner l'arbre de racine 200607 après exécution de la séquence précédente.

Utiliser la symbolique suivante:

- nom de répertoire dans un ovale,
- nom de fichier ordinaire dans un rectangle,
- nom de lien symbolique souligné et flèche en trait pointillé vers le nom de la cible,
- nom de lien physique non souligné et flèche en trait plein vers le nom de la cible.

2) – Préciser la commande (10) qui permettrait que tous les éléments de l'arbre de racine D1 aient les droits 'r-x' pour la classe d'ayants droit 'others'.

3) – Précisez la commande (11) qui permettrait de changer les droits par défaut afin que toute future création de fichier ou répertoire n'attribue aucun droit aux classes d'ayants droit 'group' et 'others'.

4) – Ecrire la formule qui fournit le nombre de liens physiques du répertoire parent du répertoire 200607. Préciser.

5) – (lire 'Annexe de I.2')

Pour l'arbre de racine '200607', écrire de façon simple mais précise, les contenus en mode symbolique, des i-nodes alloués, et les contenus en mode symbolique des blocs de données alloués. Utilisez pour ceci la dernière feuille du sujet (Table I.2.5) que vous dégraferez pour la joindre, identifiée par votre numéro d'anonymat, à votre copie d'examen.

Annexe de I.2 :

Sur la feuille, les i-nodes sont repérés par des numéros relatifs de i-nodes commençant à 100, et se substituant aux réels numéros relatifs de i-nodes.

Champs d'un i-node auxquels il s'agira de s'intéresser:

- type du fichier (-, d, l)
- nombre de liens (hard links) sur le fichier
- taille en octets
- une adresse relative (adressage direct) de bloc de données.

Les blocs de données sont repérés par des numéros relatifs de blocs de données.

Un bloc de données d'un fichier-répertoire est constitué d'entrées qui associent chacune un nom à un numéro de i-node.

Un tel bloc possède toujours dès la création, 2 entrées garnies, celles de noms '.' et '..'.

***** **Fin de l'Annexe de I.2** *****

**Unix: Etude de cas – réseau de postes avec Unix
Partitions, systèmes de fichiers, fichiers
Gestion de comptes et de 'home directories'****(sujet: 2 pages)****(70 minutes - 12 points)**

Le cadre est celui de 'Travaux sur Machines' pour une 'Unité d'Enseignement' de nom: 'Initiation à l'administration de systèmes'. Une salle est disponible avec 8 ordinateurs identiques, chacun identifié par son numéro, compris entre 1 et 8.

Chacun des postes est équipé entre autres :

- d'un disque dur de 120 Go (interface IDE, canal 'primary', 'master')
- d'un lecteur de CD (interface IDE, canal 'secondary', 'master')
- d'un lecteur de disquettes
- d'une carte de communication pour réseau Ethernet.

On dispose pour cette salle:

- d'un commutateur de type 'hub'
- des câbles nécessaires aux liaisons des ordinateurs au 'hub'
- de jeux de CD d'une version de Linux – Fedora Core 4.

La salle sera utilisée pour 4 groupes de 'TME', de chacun 8 étudiants, pendant une douzaine de séances hebdomadaires. Chaque étudiant pendant un semestre reste associé à un poste, au sein de son groupe de TME , afin de retrouver ses travaux de façon incrémentale, de séance en séance.

Sur chaque poste sont installées 4 occurrences du système Linux avec des possibilités minimum (éditeurs de textes, adressage IP fixe, outils pour un mode serveur NFS, NIS n'est pas en oeuvre). Chaque occurrence du système Linux dispose de 30 Go sur le disque.

Chaque occurrence du système Linux est installée avec 3 partitions:

- la partition système de nom '/',
- la partition de swap,
- une 'extra partition' dont le point de montage est:
 - soit '/users' (postes 1 et 2) dédiée aux 'home directories' des utilisateurs du réseau,
 - soit '/nfs_applications/Stack_of_int (poste 4) bibliothèque partageable,
 - soit '/nfs_applications/Suite_de_Fibonacci (poste 3) programmes partageables,
 - soit '/nfs_tmp' (postes 5 et 6) équivalent pour le réseau, de /tmp local,
 - soit '/nfs_ro' (postes 8 et 7) dédiée à la diffusion de documents en lecture seule.

Pour chaque groupe de TME, 2 sous réseaux de chacun 4 postes, sont construits.

Le sous réseau 1 est composé des postes 1, 4, 5 et 8.

Le sous réseau 2 est composé des postes 2, 3, 6 et 7.

Le sous réseau 1 est structuré autour de l'adresse IP fixe 192.168.1.0.

Le sous réseau 2 est structuré autour de l'adresse IP fixe 192.168.2.0.

Le 4^e octet des adresses IP fixes correspond au n^o relatif de poste.

Au sein d'un sous réseau, chaque poste rend exportable son 'extra partition' aux 3 autres postes, et monte les 'extra partitions' des 3 autres postes.

Les installations ont été faites dans l'ordre de travail dans la semaine des groupes de TME, soit: de 1 à 4.

Evolutions des postes et des sous réseaux à la 7^e semaine du semestre.

Par poste, il est possible de ne définir qu'une partition de 'swap'. On retient la première partition de 'swap' créée (celle du TME 1).

On regroupe les sous réseaux 1 et 2 en un seul sous réseau.

Le poste 1 est doté d'un second disque IDE de 120 Go pour supporter les partitions /users (une par groupe de TME).

Le poste 2 est dédié aux sauvegardes des home-directories des utilisateurs (partitions /users_backup).

Le poste 4 regroupe sous /nfs_applications, les diverses applications partageables (bibliothèques et exécutable) des postes 4 et 3. Le poste 3 ne gère plus d'extra partition.

Les postes 5 et 8 ne gèrent plus d'extra partitions. Leurs rôles sont respectivement repris par 6 et 7.

!!! Jusqu'à ce point, le contexte est le même que celui de l'examen de première session !!!

Les noms des postes sont imposés: P1 à P8.

On considère que l'administration des 'accounts' des utilisateurs et groupes d'utilisateurs Unix est locale à chaque poste avant ou après les "Evolutions".

Les noms des partitions physiques locales, utiles avant les "Evolutions", et encore utiles après les "Evolutions", sont les mêmes.

On se situe à la fin de la 6^e semaine:

- les 2 sous réseaux existent et fonctionnent correctement,
- et aucune des modifications nécessaires pour répondre aux "Evolutions" n'a encore eu lieu.

On ne s'intéressera qu'à ce qui concerne le groupe de TME 2.

Question:

1) – Ecrire le protocole précis de gestion des 'groups', des 'users' et de leurs 'home-directories' pour réaliser la transition vers l'unique sous réseau suite à ces évolutions, afin de ne rien perdre des travaux réalisés par les utilisateurs des 2 sous réseaux pendant les premières 6 semaines du semestre.

ATTENTION:

Soyez précis dans l'ordre des commandes à exécuter, leurs options, leurs arguments.

Tous les noms de fichiers ordinaires, de répertoires, de commandes, seront des noms absolus.

Il y a une quarantaine d'actions en cause. Dans le désordre:

- gérer les 4 fichiers d'administration des comptes, pour groupes et utilisateurs,
- organiser sauvegardes, transferts, restaurations des contenus des 'home-directories', (sans faire intervenir l'usage de clés USB de mémoire, et de disques gravés),
- créer / détruire des partitions de disques,
- intégrer des postes d'un sous réseau au sein d'un autre sous réseau,
- rendre exportables et monter des partitions, ...