

Sujet d'Examen Session 2 du 18 juin 2009

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

Documents autorisés, mais non transmissibles

Unix (120 minutes - 20 points)

Shell scripts – Expressions régulières
Système de fichiers – Partitions – Comptes utilisateurs
Etude de cas

Gérard Nowak

Les questions I.1, I.2, I.3 sont indépendantes.

I.1 Shell scripts – Expressions régulières (15 minutes - 4 points)

Un administrateur a produit grâce à un Shell script un fichier de nom `passwdyp` (dans lequel tout enregistrement est censé respecter la syntaxe associée au fichier `/etc/passwd`), pour la gestion de comptes de nouveaux utilisateurs.

Voir des extraits de ce fichier, ci-dessous.

L'administrateur crée un second Shell script de nom `check_passwdyp` recevant comme argument le nom du fichier à contrôler. Ce script ne contient qu'une ligne: usage de la commande `grep` à laquelle on transmet une expression régulière. Voir des extraits de '`man grep`' en Annexe.

Ce script doit produire sur son flux de sortie (fichier dont le nom: `passwdyp.erreurs` est transmis lui aussi en argument) les enregistrements du fichier contrôlé qui ne respectent pas les quelques règles suivantes, et seulement celles-ci :

- chaque enregistrement contient 7 champs séparés par le caractère ':'
- le 1^{er} champ ne peut pas être vide et contient 8 caractères au maximum dans l'alphabet composé, dans le cas présent, des lettres minuscules et du caractère '_'
- le 2^e champ ne peut pas être vide
- le 3^e champ doit correspondre aux UID compris entre 1100 et 1299, ces 2 valeurs incluses
- le 4^e champ correspond aux groupes 1100 ou 1200
- le 6^e champ commence par '/users/' et permet de noter ensuite un niveau lié au groupname (mais peu importe ce nom) et un niveau lié au username (mais peu importe ce nom)
- le 7^e champ est '/bin/bash'.

Exemples :

```
$ cat passwdyp
du_calme:x:1101:1100:Du Calme Germain:/users/g1/du_calme:/bin/bash
babour:x:1102:1100:Babour Adel:/users/g1/babour:/bin/bash
n_guyen:x:1201:1200:N'Guyen Samiha Lea:/users/g2/n_guyen:/bin/bash
herman_m:x:1202:1200:Herman-Master William:/users/g2/herman_m:/bin/bash
du_calme:x:1103:1100:Du Calme Andre:/users/g1/du_calme:/bin/bash
ba-bour:x:1104:1100:Ba-Bour Gilles:/users/g1/ba-bour:/bin/bash
nguyen::1203:1200:Nguyen Adele:/users/g2/nguyen:/bin/sh
```

```

herman_master:x:1204:1200:Herman-Master William:/users/g2/herman_master:/bin/bash
babel:1212:1200:Babel Abel:/users/g2/babel:/bin/bash
dualme:x:1121:1100:Du Calme Henri:/users/g1/dualme:/bin/bash
dubourg:x:1105:11:Dubourg Eloi:/users/g1/dubourg:/bin/bash
larousse:y:1250:1200:Larousse Anne Marie:/users/g2/larousse:/bin/bash
hermann:x:1206:1200:Hermann Harald:/user/g2/hermann:/bin/bash
guyer:x:1207:1200:Guyer Georges:/users/g2/guyer:/bin/sh
laroue:y:1350:1200:Laroue Marie Ange:/users/g2/laroue:/bin/bash
larosse:y:2101:1200:Larosse Annie:/users/g2/larosse:/bin/bash
rossa:y:1208:1300:Rossa Bella:/users/g2/rossa:/bin/bash
:y:1209:1200:Larossa Angele:/users/g2/larossa:/bin/bash
rossy:y:1210:1200:Rossy Jim:/users/rossy:/bin/bash

```

\$ cat passwdp.erreurs

```

du calme:x:1103:1100:Du Calme Andre:/users/g1/du calme:/bin/bash
ba-bour:x:1104:1100:Ba-Bour Gilles:/users/g1/ba-bour:/bin/bash
nguyen::1203:1200:Nguyen Adele:/users/g2/nguyen:/bin/sh
herman_master:x:1204:1200:Herman-Master William:/users/g2/herman_master:/bin/bash
babel:1212:1200:Babel Abel:/users/g2/babel:/bin/bash
dubourg:x:1105:11:Dubourg Eloi:/users/g1/dubourg:/bin/bash
hermann:x:1206:1200:Hermann Harald:/user/g2/hermann:/bin/bash
guyer:x:1207:1200:Guyer Georges:/users/g2/guyer:/bin/sh
laroue:y:1350:1200:Laroue Marie Ange:/users/g2/laroue:/bin/bash
larosse:y:2101:1200:Larosse Annie:/users/g2/larosse:/bin/bash
rossa:y:1208:1300:Rossa Bella:/users/g2/rossa:/bin/bash
:y:1209:1200:Larossa Angele:/users/g2/larossa:/bin/bash
rossy:y:1210:1200:Rossy Jim:/users/rossy:/bin/bash

```

Questions :

I.1.a) – Ecrire le contenu du fichier check_passwdp.

I.1.b) – Ecrire une commande permettant de le rendre exécutable pour l'administrateur seulement, de l'activer dans les conditions prévues pour son utilisation, d'afficher le code retour du script interprété, et d'afficher à l'écran le fichier produit.

I.2 Système de fichiers – Partitions – Comptes utilisateurs (60 minutes - 10 points)

Les home directories sont organisés selon 3 niveaux: users, un groupname, un username.

Un administrateur est sollicité par deux utilisateurs de usernames dujardin et lefort, de groupname g1, ayant chacun un compte (inutilisables d'après ces utilisateurs) sous Linux pour le réseau géré.

Résultats de la commande : **cat /etc/passwd (extraits)**
dujardin:x:1181:2100:Dujardin Jeanne:/users/g1/dujardin:/bin/bash
lefort:x:1192:2100:Lefort Jules:/user/g1/lefort:/bin/bash

Résultats de la commande : **ls -aRl /users/g1/dujardin**

```

drwxr-xr-x  2   root   g1   4096  ...  .
drwxr-xr-x 66   root   root  4096  ...  ..
-rw-r--r--  1  dujardin g1   ...   ...  .bashrc
-rw-r--r--  1  dujardin g1   ...   ...  .bash_profile

```

Questions : (1 point)

Au vu de ces seules informations (les '...' notent ici des informations non intéressantes) :

I.2.a) – Quel est a priori le problème pour l'utilisateur de username dujardin ?

1.2.b) – Comment précisément régler son problème ?

Ecrire les commandes et / ou mises à jour de fichier(s) nécessaires.

I.2.c) – Quel est a priori le problème pour l'utilisateur de username lefort ?

I.2.d) – Comment précisément régler son problème ?

Ecrire les commandes et / ou mises à jour de fichier(s) nécessaires.

Un troisième utilisateur a très mal mis à jour ses fichiers de commandes initiales pour bash. Il ne parvient plus à entrer en session authentifiée.

Question : (2 points)

I.2.e) – Comment peut-il, sans déranger l'administrateur, revenir à une situation normale (à nouveau pouvoir entrer en session authentifiée). Précisez toutes les étapes.

Sur un des postes du réseau géré, deux installations de Linux cohabitent. L'administrateur (débutant) ne se souvient plus du mot de passe pour 'root' de la seconde installation de Linux sur ce poste. Il se souvient très bien du mot de passe de 'root' pour la première installation de Linux. Mais les mots de passe sont différents.

Question : (3 points)

I.2.f) – Comment peut-il revenir à une situation normale (réinstaller un mot de passe pour 'root' pour la seconde installation de Linux). Précisez toutes les étapes, les détails des commandes, ... Attention : il ne s'agit pas de ré-installer Linux.

Soit un extrait du fichier /etc/fstab d'un poste :

```
/dev/hda1 / ext3 defaults 1 1
/dev/hda2 /users ext3 defaults, noauto 1 2
```

Notre administrateur (débutant) ne comprend pas les résultats suivants :

```
$ ls -aRl /users
```

```
..
drwxr-xr-x 4 root root 4096 ... .
drwxr-xr-x 12 root root 4096 ... ..
drwxr-xr-x 2 root root 4096 ... g1
drwxr-xr-x 2 root root 4096 ... g2
```

```
./g1:
```

```
drwxr-xr-x 2 root root 4096 ... .
drwxr-xr-x 4 root root 4096 ... ..
```

/g2:

```
drwxr-xr-x 2      root  root  4096 ... .  
drwxr-xr-x 4      root  root  4096 ... ..
```

Questions : (2 points)

I.2.g) – Comment expliquer que les répertoires g1 et g2 devant contenir les home directories des utilisateurs connus, soient vides ?

I.2.h) – Peut-on détruire les répertoires g1 et g2 avant de réaliser ce qui concerne la question I.2.i) ? Si votre réponse est : Oui, considérez que pour la suite les répertoires g1 et g2 auront été détruits en I.2.h).

I.2.i) – Que faire (commande précise) afin que, sans reboot, pour la suite, les home directories soient visibles dans g1 et g2 ? On considère que la commande est réalisée pour la suite. Décrire en 2 lignes ce que produira la commande suivante, avec toutes précisions pour un utilisateur:

```
ls -aRl /users/g1
```

I.2.j) – Que faire (commande précise) afin que, les home directories soient visibles dans g1 et g2, lors du prochain reboot ?

I.2.k) – Toujours dans la suite et sans reboot, et selon ce que vous aurez répondu en I.2.h), quel sera le résultat produit par la séquence:

```
umount /users  
ls -aRl /users
```

L'administrateur doit sauvegarder puis supprimer tout ce qui concerne un compte : utilisateur de username dujardin du groupe de groupname g1.

Question : (2 points)

I.2.l) – On considère maintenant que l'administration des users et des groupes est centralisée grâce à NIS sur le réseau. Précisez dans l'ordre toutes les étapes nécessaires pour réaliser la tâche de l'administrateur concernant le compte de username dujardin, de groupname g1. A chaque étape précisez la ou les commande(s), les options et arguments, les modifications précises dans les fichiers.

I.3 Unix – Etude de cas

(45 minutes - 6 points)

Vous disposez de tous les éléments (ordinateurs, disques, switch, câbles, jeux de CD pour l'installation de Linux) pour construire un réseau local constitué d'un ordinateur serveur, inaccessible aux utilisateurs, et de 15 postes réservés aux sessions des utilisateurs ordinaires et d'une imprimante réseau.

L'usage est dédié à l'apprentissage de développements en langage C.

Les utilisateurs seront au maximum une centaine tous dans un seul groupe.

Questions :

I.3.a) – En quelques phrases courtes, quelle organisation pour ce réseau local, proposez vous ? Un schéma détaillé et annoté est bienvenu.

Précisez aussi le nombre de disques durs que vous préconisez par poste, et leurs usages.

I.3.b) – Précisez les relations "d'import/export" de partitions entre postes, du point de vue de NFS, pour le réseau local que vous proposez.

I.3.c) – Quelle organisation proposez vous si vous disposez de 2 postes serveur NFS pour le réseau local ?

Annexe pour I.1

Extraits de : man grep

NAME

grep, egrep, fgrep - 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 (see below).

-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. The fundamental building blocks are the regular expressions that match a single character. Most characters, including all letters and digits, are regular expressions that match themselves. Any metacharacter with special meaning may be quoted by preceding it with a backslash.

A bracket expression is a list of characters enclosed by [and]. It matches any single character in that list; if the first character of the list is the caret ^ then it matches any character not in the list. For example, the regular expression [0123456789] matches any single digit.

Within a bracket expression, a range expression consists of two characters separated by a hyphen. It matches any single character that sorts between the two characters, inclusive, using the locale's collating sequence and character set. For example, in the default C locale, [a-d] is equivalent to [abcd].

Finally, certain named classes of characters are predefined within bracket expressions, as follows. Their names are self explanatory, and they are [:alnum:], [:alpha:], [:cntrl:], [:digit:], [:graph:], [:lower:], [:print:], [:punct:], [:space:], [:upper:], and [:xdigit:]. For example, [[:alnum:]] means [0-9A-Za-z], except the latter form depends upon the C locale and the ASCII character encoding, whereas the former is independent of locale and character set. (Note that the brackets in these class names are part of the symbolic names, and must be included in addition to the brackets delimiting the bracket list.) Most metacharacters lose their special meaning inside lists. To include

a literal] place it first in the list. Similarly, to include a literal ^ place it anywhere but first.

Finally, to include a literal - place it last.

The period . matches any single character. The symbol \w is a synonym for [[:alnum:]] and \W is a synonym for [^[:alnum:]].

The caret ^ and the dollar sign \$ are metacharacters that respectively match the empty string at the beginning and end of a line. The symbols \< and \> respectively match the empty string at the beginning and end of a word. The symbol \b matches the empty string at the edge of a word, and \B matches the empty string provided it's not at the edge of a word.

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.

Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated subexpressions.

Two regular expressions may be joined by the infix operator |; the resulting regular expression matches any string matching either subexpression.

Repetition takes precedence over concatenation, which in turn takes precedence over alternation.

A whole subexpression may be enclosed in parentheses to override these precedence rules.

The backreference \n, where n is a single digit, matches the substring previously matched by the nth parenthesized subexpression of the regular expression.

In basic regular expressions the metacharacters ?, +, {, |, (, and) lose their special meaning; instead use the backslashed versions \?, \+, \{, \|, \(, and \).