

Examen ESIAL 1A, samedi 24 avril 2004 module Langage C et programmation Shell

- Programmation Shell -

Durée : 2h. Tout type de document autorisé.

Les questions sont indépendantes. Le barème n'est donné qu'à titre indicatif.

Question 1 (3 pts) Soit un fichier `std.cshrc` qui est une version standard de `.cshrc` situé dans le répertoire /etc sous la racine.

```
# Default user .cshrc file
# Usage : copy this file to a user's home directory and edit it to
# customize it to taste. It is run by csh each time it starts up.
# set up default command search path :
set path = ( /bin /usr/bin )

# set up C shell environment :
if ($?prompt) then
  set history = 20
  set savehist = 20
  set system = $cmd:t
  set prompt = "$system \!:"
endif

# sample alias
alias h history
endif
```

On suppose que votre home directory est un sous-répertoire direct de /users/esial1 et que pour cet exercice vous vous trouvez dans votre home directory.

On suppose également que prompt est une variable prédéfinie initialisée par défaut à %

1. Ecrivez une commande permettant de copier ce fichier sous le nom `.cshrc` dans votre répertoire privé en utilisant les chemins absolus uniquement.
2. Ecrivez cette même commande avec des chemins relatifs uniquement.

1

ESIAL 1A 2003/2004 - Examen Programmation Shell

3. Comment modifier ce fichier pour que le système UNIX trouve les scripts du répertoire courant quel que soit ce répertoire ? Quels sont les dangers d'une telle modification ?

4. Comment faire pour que la commande `ls -l` puisse s'appeler par `ll` ?

5. Commentez les deux lignes suivantes :

```
set system = $cmd:t
set prompt = "$system \!:"
```

Question 2 (3 pts) Le fichier `oups` contient le script-shell suivant :

```
#!/bin/csh
setenv PATH .
```

Expliquez les deux séquences suivantes, et en particulier en quoi elles sont différentes :

```
séquence 1 :
% date
Sat Apr 24 8:11:57 NEST 2004
% chmod u+x oups
% oups
% date
Sat Apr 24 8:11:57 NEST 2004

séquence 2 :
% date
Sat Apr 24 8:11:57 NEST 2004
% source oups
% date
date: Command not found.
```

Question 3 (3 pts) Ecrivez de commandes `csh` :

1. Ecrire en `csh` la commande

```
estdans nom liste_de_noms
```

qui affiche sur la sortie standard le nombre de fois où `nom` figure dans la liste de noms.

2. Ecrire en `csh` la commande

```
enlever nom liste_de_noms
```

qui affiche sur la sortie standard la liste de noms privée de toutes les occurrences de `nom`.

Question 4 (3 pts) Ecrire une commande

```
cpdir rep1 rep2
```

qui recopie tous les fichiers de `rep1` dans `rep2` avec un message lors de chaque copie qui échoue (par exemple, tentative de copie d'un sous-répertoire), mais aussi lors de chaque copie réussie.

Page 2

Question 5 (4 pts) Ecrire une commande

groupe <liste_de_fichiers>

qui écrit sur la sortie standard, pour chacun des fichiers de liste_de_fichiers, son nom puis son contenu, selon le format suivant :

% groupe *.c

| prog1.c |

... contenu de prog1.c ...

| prog2.c |

... contenu de prog2.c ...

| progN_un_peu_plus_long.c |

... contenu de progN_un_peu_plus_long.c ...

Notes en particulier que l'encadrement de chaque nom de fichier est adapté à sa taille.

Question 6 (4 pts)

1. Ecrire en langage C la commande division permettant de calculer le quotient et le reste de la division entière de deux valeurs passées en paramètres.

Par exemple :

% division 15 4

quotient : 3

reste : 3

Le programme renverra le code de retour 2 si le nombre de paramètres est incorrect, le code de retour 1 si le deuxième paramètre est nul et le code de retour 0 sinon.

2. Ecrire le script cah_appelDiv qui, après avoir lu deux valeurs entières entrées au clavier, appelle la commande division en lui passant ces deux valeurs en paramètres. Le script affichera un message d'erreur si la commande division renvoie un code de retour strictement positif.