19. Ajoutez à la fin de votre fichier `.bashrc` une commande affichant « bashrc : Bonjour ! » puis testez-la en lançant un nouveau shell (commande `bash`) et en ouvrant une nouvelle fenêtre terminal.
20. Faites la même manipulation avec le fichier `.profile` (affichez « profile : Bonjour ! ») et observez les différences. Notez que si le message est affiché à l'ouverture d'un terminal, c'est que le terminal a été défini comme *terminal de connexion*. Ouvrez le profil par défaut dans le terminal : `Édition/Préférences...` puis cliquez sur l'onglet `Commande`, l'option `Lancer la commande en tant que shell de connexion` doit être cochée.
21. On peut lancer `bash` comme un « shell de connexion » (il exécute alors votre fichier `.profile`). Avec `man bash` déterminer quelle est l'option à utiliser pour cela. Donnez la commande :

bash -l

Compte-rendu

Vous rendrez à l'enseignant un exemplaire du sujet complété à la fin du TP. Si vous n'avez pas fini, utilisez l'exemplaire qui vous reste pour terminer pendant votre temps libre.

FAUGERON Tylan
CHALENCON Yoan
5

10/10

Utilisation du shell Unix
3 - Droits d'accès et environnement

Rappel : vous devez simplement effectuer le travail demandé et noter les commandes utilisées et les réponses aux questions dans l'espace laissé libre. **Attention :** l'objectif est que vous soyez capables de refaire SEUL les manipulations décrites.

Pré-requis

Vous devez avoir parcouru le chapitre « Le langage de commande » dans les diapos de cours : R1.04-IntroSysteme/Unix.pdf

1 Droits d'accès

Créez un répertoire `TP3` dans le répertoire `mesTPs/Systeme` (ou dans le répertoire où vous avez l'habitude de ranger vos TP d'introduction aux systèmes).
Placez-vous dans ce répertoire `TP3`.

1. Copiez le répertoire `/users/but/info/Public/DroitsAcces` dans le répertoire courant (`TP3`).

cp -r /users/but/info/Public/DroitsAcces .

2. Affichez le contenu de `DroitsAcces` en montrant les propriétés des fichiers.

ls -l DroitsAcces

3. Pour chaque fichier ou répertoire, donnez ses droits d'accès en forme symbolique et en forme octale (par exemple `rwxr-xr-x` ⇒ `755`).

bonjour → rwxr-xr-x => 755
dansleslbois → rw-----r-- => 604
dicton → r--r--r-- => 444
Hand → rwxr-xr-x => 755

4. Ajoutez au fichier `danslesbois` le droit de lecture pour les membres du groupe propriétaire.

chmod g+r DroitsAcces/danslesbois

5. Ouvrez le fichier `dicton` et essayez de le modifier avec votre éditeur favori. Expliquez ce qui se passe.

gedit DroitsAcces/dicton
On ne peut pas modifier le fichier car nous n'avons pas les droits d'écriture

6. Pouvez-vous effacer le fichier `dicton` avec la commande `rm` ? Pourquoi (alors que vous ne pouvez pas modifier le contenu du fichier) ?

rm DroitsAcces/dicton
On peut le supprimer car on a les droits d'écriture sur le dossier parent.

7. Lancez le programme `bonjour` sans paramètre puis avec le paramètre « Albert ».

```
DroitsAcces/bonjour
→ Bonjour joyeux (futur) contributable !
```

```
DroitsAcces/bonjour Albert
→ Bonjour Albert !
```

8. Si vous ouvrez le fichier `bonjour` avec votre éditeur favori vous constatez que c'est du texte. Il s'agit d'un script shell (suite de commandes). Enlevez maintenant le droit `x` sur `bonjour`.

```
gedit DroitsAcces/bonjour
chmod -x DroitsAcces/bonjour
```

9. Que se passe-t-il si vous essayez à nouveau de lancer `bonjour` ?

```
DroitsAcces/bonjour
→ Permission non accordée
```

Nous n'avons plus les droits d'exécution donc on ne peut plus lancer le fichier

10. En utilisant la notation octale (affectation absolue), donnez à `bonjour` les droits suivants : tout le monde peut le lire ou l'exécuter, seul son propriétaire peut le modifier. (rwxr-xr-x)

```
chmod 755 DroitsAcces/bonjour
```

11. Affichez le contenu du répertoire `Hand`, puis supprimez pour tout le monde le droit de lecture sur ce répertoire.

```
ls DroitsAcces/Hand
chmod -r DroitsAcces/Hand
```

12. Si vous tentez maintenant d'afficher le contenu, vous obtenez un message d'erreur. Pouvez-vous quand même afficher le contenu de `Archives` dans `Hand` ? Donnez la commande et expliquez.

```
ls DroitsAcces/Hand/Archives
```

Ça fonctionne car on a les droits d'exécution sur le dossier parent (Hand) donc on peut traverser le dossier pour lire le sous-dossier Archives

2 Environnement et variables

Tout processus Unix s'exécute dans un certain *environnement*. Cet environnement contient un ensemble de variables. Une variable est l'association d'un nom et d'une valeur. Nom et valeur sont des chaînes de caractères.

On peut afficher les variables de l'environnement du shell avec la commande `set`. Testez cette commande. Le résultat n'est pas très lisible. . ..

Une partie de l'environnement est qualifiée de *publique* car elle est automatiquement transmise aux sous-processus. Dans le cas du shell cela signifie que l'environnement public est transmis à toutes les commandes que vous lancez depuis le shell. On peut afficher l'environnement public avec la commande `env`. Testez cette commande.

Pour afficher la valeur d'une variable, on peut utiliser simplement la commande `echo` du shell en faisant précéder le nom de la variable d'un `'$'`. Essayez :

```
echo $HOME
```

Pour créer une nouvelle variable ou modifier la valeur d'une variable existante, on écrira :


```
NOM=valeur
```

Attention : il ne faut pas mettre d'espaces autour du `'='`
Par défaut une variable n'est pas publique. On peut la rendre publique avec `export NOMVAR`.

13. Faites afficher l'environnement **public** de votre shell. Quelle sont les valeurs des variables `HOME`, `LOGNAME` et `SHELL` ?

```
HOME → /users/but1/info/chalencq
LOGNAME → chalencq
SHELL → /bin/bash
```

14. `PS1` est le *prompt*, affiché par le shell quand il est prêt à lire une commande. Donnez la commande permettant d'afficher la valeur de la variable `PS1` puis modifiez cette valeur.

```
echo $PS1
PS1=1
```

Pour rétablir l'ancienne valeur, il suffit de fermer la fenêtre et d'en ouvrir une nouvelle.

15. Définissez une variable `Binome` contenant vos deux noms. Vérifiez en affichant la valeur.

```
Binome=TylanYoan
echo $Binome
```

16. Copiez le fichier `/users/but/info/Public/afficheVar.c` dans votre répertoire de TP puis compilez-le.

```
cp /users/but/info/Public/afficheVar.c .
gcc afficheVar.c -o afficheVar
```

17. Lancez l'exécutable : il affiche la variable `HOME`, puis la variable `Binome`. Que constatez-vous pour `Binome` ? Expliquez.

```
./afficheVar
```

On nous dit que la variable Binome n'existe. C'est car elle n'est pas publique.

18. Rendez publique la variable `Binome` (`export Binome`) puis relancez `afficheVar` pour vérifier que vous voyez bien la variable `Binome` dans le sous-processus `afficheVar`, puisqu'elle est maintenant publique.

3 Adaptons notre environnement

L'environnement utilisateur peut être adapté en ajoutant des commandes dans deux fichiers situés dans votre répertoire *HOME directory* :

— `.profile` (parfois `.bash_profile`), exécuté à la connexion ou à l'ouverture de session ;

— `.bashrc`, exécuté à chaque fois qu'un shell est lancé.

18. Donnez la commande qui affiche les propriétés des deux fichiers `.profile` et `.bashrc` (sans changer de répertoire, utilisez la variable d'environnment `HOME`).

```
ls -l $HOME/.profile $HOME/.bashrc
```