

Conception d'un shell

Vous devez concevoir un shell rudimentaire. Votre shell doit être développé en langage C et il devra compiler et s'exécuter sur le serveur malt. Uniquement les bibliothèques disponibles sur malt sont permises.

Il ne s'agit pas d'une simulation. Les commandes de votre shell doivent avoir un réel impact sur votre machine.

Le Shell

Le nom de l'exécutable sera "tsh" pour Turbo-Shell. Votre shell aura la responsabilité de lancer divers logiciels pour réaliser des tâches simples.

Le shell doit offrir cet invite de commande pour indiquer qu'il est prêt à recevoir une nouvelle tâche :

```
tsh>
```

Le shell doit ensuite interpréter la commande et l'exécuter. Le résultat de l'exécution du logiciel appelé doit être affiché à l'écran. Lorsque la commande est terminée, l'invite de commande doit réapparaître, comme le montre l'exemple suivant :

```
tsh> list -d
Sous-répertoires de /usr/kc932134/inf3172
tp1
tp2
notes
test
tsh>
```

Si aucune commande n'est spécifiée et qu'on entre la touche Entrée, on réaffiche l'invite.

```
tsh>
tsh>
```

Le shell doit supporter directement 3 commandes : cd, cdir et exit.

cd <repertoire destination>

Comme sous Unix, la commande `cd` sert à changer le répertoire de travail du shell. Cette commande reçoit en paramètre le chemin relatif ou absolu vers le répertoire cible. La commande `cd` peut également recevoir `..` pour remonter d'un niveau dans la hiérarchie de répertoires. La commande "`cd ..`" appliquée sur le répertoire racine n'a aucun effet et n'affiche aucun message d'erreur. Si le répertoire cible n'existe pas, un message d'erreur est affiché.

```
tsh> cd test
tsh> cd /inexistant
Répertoire introuvable
tsh>
```

cdir

La commande `cdir` ne prend aucun paramètre. Elle affiche le chemin absolu du répertoire courant du shell.

```
tsh> cdir
Répertoire courant : /usr/kc932134/inf3172
tsh>
```

exit

La commande `exit` ne prend aucun paramètre. Elle met fin à l'exécution du shell.

```
tsh> exit
```

Autres commandes

Pour toute autre commande, le shell doit vérifier si l'exécutable sollicité se trouve dans un répertoire en particulier. Toutes les commandes du shell seront dans le même répertoire, c'est-à-dire `~/inf3172/bin`

Si un programme demandé n'est pas dans ce répertoire, un message d'erreur est affiché.

```
tsh> biz
Programme introuvable
tsh>
```

Les autres exécutables

Toutes ces commandes doivent s'exécuter dans un nouveau processus. Le `stdout` du nouveau processus doit être affiché par le shell.

new <nom de fichier>

La commande `new` sert à créer un nouveau fichier texte vide. Le paramètre est un chemin relatif ou absolu vers le fichier à créer. Une erreur de création doit afficher un message à l'écran.

```
tsh> new inf3172/tp1/source.c
Fichier créé
tsh> new inf3172/tp1/source.c
Impossible de créer le fichier
tsh>
```

list <-d>

La commande list sert à afficher la liste des fichiers ou répertoires dans le répertoire courant du shell. Sans l'option -d, on affiche les fichiers. Avec l'option -d, on affiche les répertoires. Les fichiers et répertoires cachés sont également affichés, mais on n'affiche pas les répertoires "." et "..".

```
tsh> list
Fichiers de /usr/kc932134/inf3172
tp1.c
tp1.h
makefile
tsh>

tsh> list -d
Sous-répertoires de /usr/kc932134/inf3172
tp1
tp2
notes
test
tsh>
```

rmall <nom du répertoire>

La commande rmall supprime un répertoire et tout son contenu récursivement (tous les sous-répertoires) sans demander de confirmation. Le nom du répertoire peut être un chemin relatif ou absolu. Si le répertoire est inexistant, un message doit être affiché. Si l'utilisateur ne possède pas les droits pour supprimer le répertoire, un message doit être affiché mais le contenu doit rester intacte.

```
tsh> rmall inf3172
Répertoire supprimé
tsh> rmall inf3172
Répertoire introuvable
tsh> rmall /dev
Permissions insuffisantes
tsh>
```

mkdir <nom du répertoire>

La commande mkdir sert à créer un nouveau répertoire vide. Le paramètre est un chemin relatif ou absolu vers le répertoire à créer. Une erreur de création doit afficher un message à l'écran.

```
tsh> mkdir inf3172/tp1
Répertoire créé
tsh> mkdir inf3172/tp1
Impossible de créer le répertoire
tsh>
```

size <nom du répertoire>

La commande size doit calculer le nombre de fichier dans le répertoire et la taille du répertoire , c'est-à-dire la somme de la taille de tous les fichiers qu'il contient, peu importe la profondeur. Le paramètre est un chemin relatif ou absolu vers le répertoire où calculer la taille. Un message doit être affiché si le répertoire est inexistant.

```
tsh> size inf3172
Le répertoire /usr/kc932134/inf3172 contient 32 fichiers pour un
total de 4578845 octets.
tsh> size temp
Répertoire introuvable
```

fin <nbLignes> <fichier>

La commande fin sert à afficher un certain nombre de lignes d'un fichier à partir de la fin du fichier. Le paramètre nbLignes est le nombre de lignes à afficher, ce doit être un entier positif. Le paramètre fichier est un chemin relatif ou absolu vers le fichier à lire. Une erreur est affichée si le fichier est inexistant.

```
tsh> fin 5 inf3172/tp1/source.c
Fichier introuvable
tsh> fin 3 inf3172/tp1/sourceTP1.c
// fin normale du programme
exit(0);
}
tsh> fin test fichier.txt
Argument invalide
tsh>
```

Gestion des erreurs

L'affichage des messages d'erreurs doit être fait à l'aide des fonctions *perror* et *strerror*.

Informations complémentaires

1. Lors de la correction, aucun caractère d'espacement ne sera utilisé dans les noms de fichiers et de répertoires.
2. Aucun lien symbolique ne sera utilisé lors de la correction.
3. Pour les commandes avec un parcours récursif de sous-répertoires, vous n'êtes pas tenus de considérer les répertoires pour lesquels vous n'avez pas les permissions d'accès. Néanmoins, ce cas ne sera pas couvert de la correction.
4. L'exécutable tsh doit être dans le même répertoire que les autres exécutables demandés.
5. Le makefile doit créer le répertoire inf3172/bin s'il n'existe pas déjà.

Remise

Le travail peut être fait seul ou en équipe de 2 personnes. Le répertoire de travail contenant les fichiers doit être archivé dans un fichier zip et nommé selon les codes permanents des auteurs. L'archive doit être remise par Moodle avant le 21 avril 2017 à 9h30. Aucun retard ne sera accepté et une pénalité sera appliquée pour une archive non conforme sans les codes permanents.

Si vous voulez travailler en équipe, vous devez mettre votre logiciel sous gestion de sources dans un dépôt privé git (ex. github, bitbucket) et m'en donner un accès en lecture. Cette mesure sert à

m'assurer que les deux coéquipiers ont partagé l'effort équitablement. Un coéquipier n'ayant pas offert une participation significative n'aura pas les points pour le travail.

Un makefile doit permettre de faire l'installation complète du logiciel à partir des sources.

Pondération

Fonctionnalité : 60%

Gestion du shell (répertoire de travail, parsing, création des processus enfants) : 25%

Qualité du code (indentation, nomenclature, absence de duplication de code) : 10%

makefile : 5%