

Ce projet a pour objectif de créer un serveur d'archive. Vous devez créer une nouvelle commande permettant de communiquer avec ce serveur.

## 1 DESCRIPTION DU PROJET

Vous allez interroger votre serveur par l'intermédiaire d'une nouvelle commande shell, nommée *vsh*, qui fonctionne selon les modes suivants:

1<sup>er</sup> mode: c'est le mode list :

```
vsh -list nom_serveur port
```

Cette commande permet d'afficher la liste des archives présentes sur le serveur. *nom\_serveur* représente l'adresse du serveur et *port* le numéro du port sur lequel le serveur attend une requête.

2<sup>ème</sup> mode: c'est le mode create :

```
vsh -create nom_serveur port nom_archive
```

Cette commande permet de créer une archive *nom\_archive* de l'arborescence contenue dans le répertoire courant de la machine cliente.

3<sup>ème</sup> mode: c'est le mode browse :

```
vsh -browse nom_serveur port nom_archive
```

Cette commande vous permet d'explorer sur le serveur l'archive *nom\_archive* en vous faisant entrer dans un mode shell *vsh*.

4<sup>ème</sup> mode: c'est le mode extract :

```
vsh -extract nom_serveur port nom_archive
```

Cette commande permet d'extraire le contenu de l'archive *nom\_archive* dans le répertoire courant de la machine cliente.

### 1.1 Description de l'archive

Une archive est un fichier qui permet de représenter l'arborescence d'un répertoire et le contenu de tous les fichiers de cette arborescence. Une archive se compose de deux parties. La première partie, appelée *header*, décrit l'arborescence des fichiers. La seconde partie, appelée *body*, représente le contenu des différents fichiers.

La première ligne de l'archive correspond à:

<numéro de ligne du début du header>:<numéro de ligne du début du body>

Puis on trouve les informations du *header* et enfin le *body*.

#### Header

Le *header* est un ensemble de répertoires juxtaposés les uns en dessous des autres.

Chaque répertoire satisfait le format suivant :

directory <dir>

Liste des fichiers et répertoires contenus dans <dir>

@

où <dir> correspond au nom du répertoire.

La liste des fichiers et répertoires contenus dans <dir> doit avoir le format suivant:

<nom> <droits d'accès> <taille> (informations complémentaires)

Par exemple, on pourra avoir:

```
toto drwxr-x--x 512
```

```
tutu -rw-r--r-- 1024 4 10
```

La première ligne décrit un répertoire qui a pour nom *toto*, pour taille 512 et pour droits d'accès *rw-r--r--*.

La seconde ligne correspond à un fichier de nom *tutu*, de taille 1024 et de droits d'accès *rw-r--r--*. Les deux nombres 4 et 10 précisent que le contenu du fichier *tutu* commence à la 4<sup>ème</sup> ligne du *body* de l'archive et occupe au total 10 lignes. On note qu'un fichier vide a pour taille 0 et ne possède donc pas d'informations complémentaires.

#### Body

Le *body* contient les contenus de tous les fichiers non vides de l'arborescence décrite dans la partie *header*. Pour simplifier, on ne traite que des fichiers textes.

#### Exemple d'archive (arch)

3:25

directory Exemple\Test\

A drwxr-xr-x 4096

B drwxr-xr-x 4096

toto1 -rwxr-xr-x 29 1 3

toto2 -rw-r--r-- 249 4 10

@

directory Exemple\Test\A

A1 drwxr-xr-x 4096

A2 drwxr-xr-x 4096

A3 drwxr-xr-x 4096

toto3 -rw-r--r-- 121 14 3

@

directory Exemple\Test\A\A1

toto4 -rw-r--r-- 0 17 0

@

directory Exemple\Test\A\A2

@

directory Exemple\Test\A\A3

@

directory Exemple\Test\B

bar -rw-r--r-- 202 17 6

@

#!/bin/bash

echo "bonjour!"

NAME

ls - list directory contents

## SYNOPSIS

ls [OPTION]... [FILE]...

## DESCRIPTION

List information about the FILES.

## DESCRIPTION

man formats and displays the on-line manual pages.

## NAME

cat - concatenate files and print on the standard output

## SYNOPSIS

cat [OPTION] [FILE]...

## DESCRIPTION

Concatenate FILE(s), or standard input, to standard output.

## 1.2 Description du mode list

Le mode list permet simplement d'afficher, sur la machine cliente, la liste des archives présentes sur le serveur. Pour simplifier le projet, on considérera que l'on dispose de l'outil qui permet de créer les archives à partir d'un système de fichiers (vous pouvez créer dans un premier temps une archive à la main à partir de l'exemple proposé au paragraphe 1.1).

## 1.3 Description du mode create

Le mode create a pour action de créer l'archive contenant toute l'arborescence des répertoires et des fichiers contenus dans le répertoire courant de la machine cliente.

La structure de l'archive obtenue avec le mode create est définie dans le paragraphe 1.1.

## 1.4 Description du mode browse

Pour implémenter les fonctions du mode browse vous aurez besoin d'une archive. Pour créer l'archive reportez vous au chapitre précédent. Nous allons utiliser l'exemple précédent pour décrire le mode browse. Pour cela nous considérerons que l'exemple est stocké dans un fichier texte noté « arch ».

Concentrez vous d'abord sur les commandes: pwd, ls et cd.

### La commande pwd:

Elle a les mêmes fonctionnalités que la fonction classique pwd : elle affiche le répertoire courant. Lorsque vous entrez dans le shell vsh, pwd doit retourner la racine c'est à dire \. Evidemment lorsque vous vous déplacerez dans l'archive à l'aide de la fonction cd, pwd devra à tout moment indiquer le répertoire courant. Par exemple,

```
$ vsh -browse arch
vsh:> pwd
\
vsh:> cd A
vsh:> pwd
```

\A

Note: Remarquez que la racine \ correspond à Exemple\Test\ dans l'archive. ATTENTION : la racine \ n'est pas la même que le root du système linux « / ».

### La commande ls:

La commande ls s'inspire elle aussi de la commande shell ls que vous connaissez, c'est-à-dire, elle liste tous les répertoires et fichiers contenus dans le répertoire courant.

```
$ vsh -browse arch
vsh:> ls
A\ B\ toto1* toto2
vsh:> ls A
A1\ A2\ A3\ toto3
```

A l'affichage, les répertoires seront suivis d'un \ et les fichiers exécutables d'une \*.

L'option « -l » permet d'afficher les informations détaillées des fichiers. Par exemple la commande ls -l A renvoie les informations suivantes :

```
drwxr-xr-x 4096 A1
drwxr-xr-x 4096 A2
drwxr-xr-x 4096 A3
-rw-r--r-- 121 toto3
```

L'option « -a » permet d'afficher les fichiers cachés. Les options combinées « -al » permettent d'afficher les informations détaillées des fichiers cachés.

D'autres options et combinaisons d'options sont possibles pour cette commande.

### La commande cd:

La commande cd permet de vous déplacer dans l'archive. Ainsi "cd \" vous permet d'aller à la racine de l'archive; "cd .." vous permet de remonter d'un niveau dans la hiérarchie; et "cd A" de vous déplacer dans le répertoire A.

### La commande cat:

La commande "cat toto1 totot2" affiche le contenu des fichiers toto1 et toto2 s'ils existent.

### La commande rm:

La commande "rm toto1" permet de supprimer le fichier toto1 de l'archive. Cette commande peut également s'appliquer à un répertoire. En conséquence, elle supprimera l'ensemble du contenu du répertoire.

Les commandes devront fonctionner avec les chemins relatifs et absolus. N'oubliez pas de traiter les messages d'erreur. Par exemple, "cd toto" doit échouer si toto est un fichier ou s'il

n'existe pas. De façon générale, vos commandes devront avoir le même comportement que celles du shell classique.

#### La commande touch:

La commande « touch toto1 » permet de créer le fichier vide toto1 si ce dernier n'existe pas.

#### La commande mkdir:

La commande « mkdir A » permet de créer le répertoire A. La commande avec l'option -p permet de créer une arborescence de répertoires de façon récursive. Par exemple mkdir -p A/B/C permet de créer le répertoire A, puis B dans A, puis le répertoire C dans B.

### **1.5 Description du mode extract**

Le mode extract a pour action de créer dans le répertoire courant toute l'arborescence de répertoires et les fichiers contenus dans l'archive nom\_archive.

Ainsi "vsh -extract nom\_serveur port nom\_archive" a pour effet de restaurer l'arborescence et les fichiers dans le répertoire courant de la machine cliente.

Vous devrez vous assurer que les répertoires et les fichiers créés ont bien les mêmes droits d'accès que ceux précisés dans l'archive. Ainsi si un fichier toto1 de l'archive a les droits -rw-r--r-- et un autre fichier toto2 a les droits -rw-----, une fois restaurés, les fichiers toto1 et toto2 auront respectivement les droits -rw-r--r-- et -rw-----.

## **2 RAPPORTS ET PRESENTATION**

Ce projet doit être effectué en binôme. Etant donné que ce projet comporte plusieurs parties, il est conseillé dès le début de se répartir le travail.

Un premier rapport de quelques pages doit être envoyé avant le 06 décembre 2021 à l'adresse [florent.retraint@utt.fr](mailto:florent.retraint@utt.fr). Ce premier rapport doit décrire la structure envisagée du programme et les choix particuliers envisagés.

Un rapport final de quelques pages doit être rendu avant la fin de la semaine précédant les finaux. Une présentation et un test du shell seront effectués pendant 15 à 20 minutes quelques jours avant l'examen final de l'UE. Bon courage !!!