



LO14 - PROJET Linux

Création d'un serveur d'archive

Alexandre Schmidt

Nathan Soufflet

Printemps 2018

Sommaire

Introduction	3
Etablissement d'une connexion	4
Le mode list	5
Le mode browse	6
Le mode extract	9

Introduction

Le but de ce projet est de réaliser un serveur permettant de stocker des archives, fichiers contenant la représentation textuelle d'une arborescence de fichiers, et de permettre à un client de naviguer dans ces archives, les modifier et les extraire sur son ordinateur.

Pour faciliter la réalisation du projet en duo, nous avons utilisé Git. Notre code est donc disponible sur <https://github.com/nathsou/LO14>

Etablissement d'une connexion

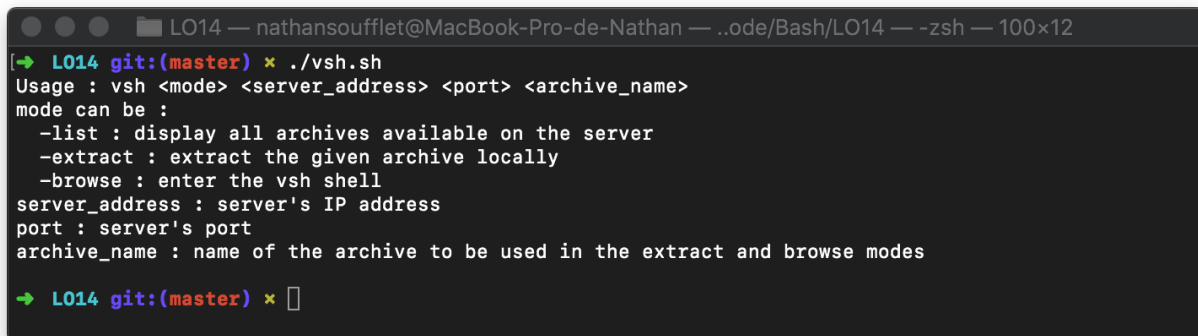
La communication entre le client et le serveur est établie grâce à la commande netcat, La sortie standard du serveur est redirigée vers une fifo qui l'envoie au client. le port est fourni en argument.

Le chemin vers le dossier des archives est fixé à "archives/" et n'est donc pas à fournir.

A terminal window titled 'LO14' with the command prompt './server.sh 1234'. The user has entered './server.sh 1234' and the output is 'Listening on port 1234'.

```
LO14 — ./server.sh 1234 — ./server.sh — ssh nathsou@192.168.0.10 -p 1789 — 100x12
[→ L014 git:(master) * ./server.sh 1234
Listening on port 1234
```

Côté client, le mode, l'adresse IP du serveur d'archive, le port utilisé ainsi que le nom de l'archive doivent être donnés en paramètre :

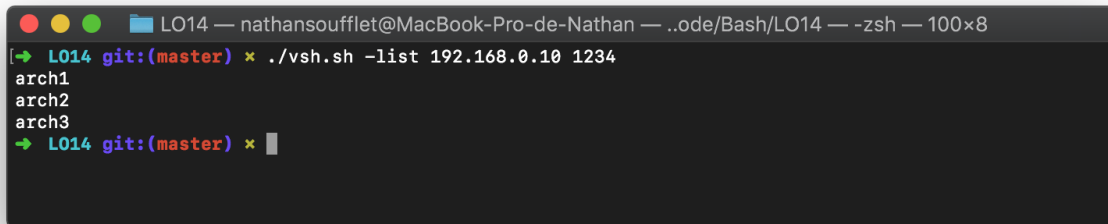
A terminal window titled 'LO14' with the command prompt 'nathansoufflet@MacBook-Pro-de-Nathan'. The user has entered './vsh.sh' and the output shows the usage and options for the vsh command.

```
LO14 — nathansoufflet@MacBook-Pro-de-Nathan — ..ode/Bash/LO14 — -zsh — 100x12
[→ L014 git:(master) * ./vsh.sh
Usage : vsh <mode> <server_address> <port> <archive_name>
mode can be :
  -list : display all archives available on the server
  -extract : extract the given archive locally
  -browse : enter the vsh shell
server_address : server's IP address
port : server's port
archive_name : name of the archive to be used in the extract and browse modes
[→ L014 git:(master) * ]
```

Lors de la connexion au serveur, le client vérifie grâce aux options "-zv" de netcat si l'adresse et le port sont en écoute.

Le mode list

Pour connaître la liste des archives disponibles sur le serveur, il suffit d'utiliser le mode `-list` :



```
LO14 — nathansoufflet@MacBook-Pro-de-Nathan — ..ode/Bash/LO14 — -zsh — 100x8
[→ LO14 git:(master) ✱ ./vsh.sh -list 192.168.0.10 1234 ]
arch1
arch2
arch3
→ LO14 git:(master) ✱
```

L'une de ces archives peut alors être utilisée comme argument des modes `extract` et `browse`.

Le mode browse

Le mode browse de ce serveur d'archive permet au client de naviguer dans l'archive sélectionnée de la même manière qu'il utiliserait le terminal pour naviguer dans l'arborescence de son ordinateur. Les commandes `pwd`, `ls`, `cd`, `cat` et `rm` ont été recréées pour pouvoir supporter la lecture d'une archive et faciliter l'utilisation du serveur par l'utilisateur.

`pwd`

La commande `pwd` permet d'afficher le chemin absolu du dossier de travail.



```
LO14 — ./vsh.sh -browse 192.168.0.10 1234 arch1 — ./vsh.sh — nc - vsh.sh -browse 192.168.0.10 123...
vsh> pwd
/A/A1/
vsh> █
```

`ls`

La commande `ls`, si elle est utilisée sans argument, affiche les dossiers et fichiers dans répertoire où se trouve l'utilisateur.

Si elle est complétée par le nom d'un dossier de l'archive (s'il est disponible, c'est-à-dire si le répertoire actuel est compris dans son chemin absolu) ou le chemin absolu de ce dossier, `ls` affiche alors le contenu de ce dossier.

Pour fonctionner, `ls` commence par vérifier si l'argument donné est cohérent, puis récupère les lignes entre le chemin absolu de l'argument et le premier `@` trouvé dans le header de l'archive.

Ensuite, les dossiers, fichiers et exécutables trouvés sont triés et affichés avec le bon signe (une étoile pour les exécutables par exemple).

```
LO14 — ./vsh.sh -browse 192.168.0.10 1234 arch1 — ./vsh.sh — nc • vsh.sh -browse 192.168.0.10 123...
vsh> ls
A/ B/ toto1* toto2
vsh>
```

cd

La commande `cd` permet de se déplacer dans l'archive. Pour cela, elle accepte plusieurs arguments :

- `/` permet de se déplacer dans la racine de l'archive
- `..` permet de se déplacer dans le dossier précédent de l'arborescence
- `<chemin absolu>` permet de se déplacer n'importe où dans l'archive
- `<chemin relatif>` permet de se déplacer dans un dossier plus profond s'il est disponible.

Le fonctionnement de `cd` est simple : la commande actualise la variable représentant le chemin absolu actuel en fonction de l'argument donné, s'il est cohérent.

```
LO14 — ./vsh.sh -browse 192.168.0.10 1234 arch1 — ./vsh.sh — nc • vsh.sh -browse 192.168.0.10 123...
vsh> ls
A1/ A2/ A3/ toto3
vsh> cd A1
vsh> pwd
/A/A1/
vsh>
```

cat

La commande `cat` permet d'afficher le contenu du fichier donné en argument.

Si le fichier existe, qu'il est disponible et non vide, `cat` récupère le début du body, le début et la fin du contenu du fichier et affiche les lignes correspondantes dans l'archive.

```
LO14 — ./vsh.sh -browse 192.168.0.10 1234 arch1 — ./vsh.sh — nc • vsh.sh -browse 192.168.0.10 123...
vsh> ls
A/ B/ toto2
vsh> cat toto2
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
```

rm

La commande `rm` permet de supprimer le fichier ou dossier (avec son contenu) donné en argument.

A terminal window with a dark background and light text. The window title bar shows 'LO14' and some connection details. The terminal content shows a series of commands and their outputs: 'vsh> ls' returns 'A1/ A2/ A3/ toto3'; 'vsh> rm toto3' returns 'Suppression du fichier Exemple/Test/A/toto3' and 'Suppression de 3 lignes'; 'vsh> ls' returns 'A1/ A2/ A3/'; and 'vsh>' is followed by a cursor.

```
LO14 — ./vsh.sh -browse 192.168.0.10 1234 arch1 — ./vsh.sh — nc - vsh.sh -browse 192.168.0.10 123...  
vsh> ls  
A1/ A2/ A3/ toto3  
vsh> rm toto3  
Suppression du fichier Exemple/Test/A/toto3  
Suppression de 3 lignes  
vsh> ls  
A1/ A2/ A3/  
vsh> █
```

Elle commence par vérifier si l'argument est un dossier ou un fichier, puis se place dans le sous-dossier le plus profond s'il y en a un grâce à une fonction récursive. Ensuite, `rm` supprime d'abord les fichiers présents dans ce sous-dossier, puis le sous-dossier lui-même jusqu'à être revenue dans le dossier placé en argument qui est supprimé à son tour.

clear

La commande `clear` efface l'historique de la console.

Le mode extract

Plusieurs solutions ont été envisagées afin d'extraire un fichier du serveur sur la machine du client, comme la conception d'un protocole maître-client où le serveur envoie des commandes atomiques les unes à la suite des autres. Une solution plus simple et plus robuste a toutefois été retenue : Le client dispose en local du script "extract.sh" permettant d'extraire directement une archive. Le serveur envoie alors le contenu de l'archive demandée et celle-ci est sauvegardée temporairement en local, le temps de l'extraction.

Par défaut, l'archive est extraite dans le dossier de travail, un autre chemin peut être indiqué en paramètre.

```
./vsh.sh -extract localhost 1234 arch1 out
```

▼	out	Today at 01:07	--	Folder
▼	Exemple	Today at 01:08	--	Folder
▼	Test	Today at 01:08	--	Folder
▼	A	Today at 01:08	--	Folder
▼	A1	Today at 01:07	--	Folder
	toto4	Today at 01:07	Zero bytes	TextEdit
▼	A2	Today at 01:07	--	Folder
▼	A3	Today at 01:07	--	Folder
	toto3	Today at 01:07	63 bytes	TextEdit
▼	B	Today at 01:07	--	Folder
	bar	Today at 01:07	107 bytes	TextEdit
	toto1	Today at 01:07	29 bytes	Unix executable
	toto2	Today at 01:07	179 bytes	TextEdit