

42sh

Interpréteur de commandes

```
jabbar_y ~/home/jabbar_y/rendu/42sh env
CPLUS_INCLUDE_PATH=/home/jabbar_y/.froot/include/
XDG_VTNR=7
LC_PAPER=fr_FR.UTF-8
LC_ADDRESS=fr_FR.UTF-8
XDG_SESSION_ID=c2
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/jabbar_y
LC_MONETARY=fr_FR.UTF-8
CLUTTER_IM_MODULE=xim
GPG_AGENT_INFO=/home/jabbar_y/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
WINDOWID=75500032
LC_NUMERIC=fr_FR.UTF-8
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1792
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=jabbar_y
LD_LIBRARY_PATH=/home/jabbar_y/.froot/lib/
```

SOMMAIRE

1. Présentation du programme

2. Lancement du programme

3. Comment fonctionne t-il ?

3.1 Fonctions de base

3.2 Fonctions avancées

1. PRÉSENTATION DU PROGRAMME

Le programme est un interpréteur de commandes UNIX.

Il a été créé sous l'impulsion d'un Projet EPITECH, nommé 42sh.

Les créateurs du programme sont :

- Rémi GASTALDI
- Renaud JULIANI
- Yassir JABBARI
- Matthias PROST
- Antoine ROIG

2. LANCEMENT DU PROGRAMME

Pour lancer le programme, il faut saisir la commande « ./42sh ».

Un prompt apparaît, vous pouvez lancer des commandes.

Ex :

```
jabbar_y@jabbari:~/rendu/42sh$ ./42sh
jabbar_y ~/home/jabbar_y/rendu/42sh ls
42sh a.out include Makefile moul.sh sources tests
jabbar_y ~/home/jabbar_y/rendu/42sh _
```

La commande « ls » permet d'afficher le contenu de votre dossier courant.

3. COMMENT FONCTIONNE T-IL ?

Le prompt est composé de votre nom et du dossier courant.

Il y a des multitudes de commandes, pour les voir utilisez la commande « man man ».

La majorité des commandes ont des options qu'on peut utiliser.

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh ls -l
total 264
-rwxrwxr-x 1 jabbar_y jabbar_y 230312 juin 6 15:13 42sh
-rwxrwxr-x 1 jabbar_y jabbar_y 6608 juin 6 12:18 a.out
drwxrwxr-x 2 jabbar_y jabbar_y 4096 juin 6 15:11 include
-rw-rw-r-- 1 jabbar_y jabbar_y 2167 juin 6 13:34 Makefile
-rwxrwxr-x 1 jabbar_y jabbar_y 4529 juin 6 12:18 moul.sh
drwxrwxr-x 2 jabbar_y jabbar_y 4096 juin 6 15:13 sources
-rw-rw-r-- 1 jabbar_y jabbar_y 5206 juin 6 12:18 tests
jabbar_y ~/home/jabbar_y/rendu/42sh _
```

La commande « ls -l » permet d'afficher le contenu du fichier courant sous forme de liste.

3.1 FONCTIONS DE BASE

REDIRECTIONS :

Celles-ci permettent de rediriger le contenu d'un résultat d'une commande vers une commande, ou un fichier.

REDIRECTION GAUCHE :

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh cat < fichier  
Wesh la famille bien ?
```

La commande « cat < fichier ».

Le contenu du fichier est redirigé vers la commande « cat » qui affiche le contenu du fichier.

REDIRECTION DROITE :

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh ls > fichier
jabbar_y ~/home/jabbar_y/rendu/42sh cat fichier
42sh
a.out
fichier
include
Makefile
moul.sh
sources
tests
```

La commande « ls > fichier »

Cette commande, envoie le résultat de la commande « ls » dans le fichier.

REDIRECTION DOUBLE DROITES :

Ex:

```
jabbar_y ~/home/jabbar_y/rendu/42sh ls -l >> fichier
jabbar_y ~/home/jabbar_y/rendu/42sh cat fichier
total 268
-rwxrwxr-x 1 jabbar_y jabbar_y 230312 juin 6 15:13 42sh
-rwxrwxr-x 1 jabbar_y jabbar_y 6608 juin 6 12:18 a.out
-rw-rw-r-- 1 jabbar_y jabbar_y 0 juin 6 15:27 fichier
-rw-rw-r-- 1 jabbar_y jabbar_y 25 juin 6 15:25 fin
drwxrwxr-x 2 jabbar_y jabbar_y 4096 juin 6 15:11 include
-rw-rw-r-- 1 jabbar_y jabbar_y 2167 juin 6 13:34 Makefile
-rwxrwxr-x 1 jabbar_y jabbar_y 4529 juin 6 12:18 moul.sh
drwxrwxr-x 2 jabbar_y jabbar_y 4096 juin 6 15:13 sources
-rw-rw-r-- 1 jabbar_y jabbar_y 5206 juin 6 12:18 tests
```

La commande « ls -l >> fichier »

Elle permet d'envoyer le résultat de « ls -l » à la fin de « fichier »

REDIRECTIONS DOUBLE GAUCHES :

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh cat << fin
? Wesh
? bien
? fin
Wesh
bien
```

La commande « cat << fin »

Un deuxième prompt apparaît et à chaque fois qu'on tape quelque chose et que nous appuyons sur « entrée », il est envoyé dans la commande « cat » jusqu'à ce qu'on tape « fin ».

Pipe :

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh ls | cat -e
42sh$
a.out$
fin$
include$
Makefile$
moul.sh$
sources$
tests$
```

La commande « `ls | cat -e` »

La commande « `ls` » envoie le contenu du dossier courant vers la commande « `cat` » avec l'option « `-e` » qui affiche à chaque fin de ligne un « `$` ».

Nous pouvons assembler les pipes, des redirections à l'infini.

Builtins :

Les builtins sont des commandes spécifiques à l'interpréteur de commande.

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh env
CPLUS_INCLUDE_PATH=/home/jabbar_y/.froot/include/
XDG_VTNR=7
LC_PAPER=fr_FR.UTF-8
LC_ADDRESS=fr_FR.UTF-8
XDG_SESSION_ID=c2
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/jabbar_y
LC_MONETARY=fr_FR.UTF-8
CLUTTER_IM_MODULE=xim
GPG_AGENT_INFO=/home/jabbar_y/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
WINDOWID=75500032
LC_NUMERIC=fr_FR.UTF-8
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1792
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=jabbar_y
LD_LIBRARY_PATH=/home/jabbar_y/.froot/lib/
```

La commande « env »

Elle affiche l'environnement de fonctionnement du programme. Cet environnement permet le fonctionnement du programme, sans celle-ci, impossible d'utiliser une commande.

Ex:

```
jabbar_y ~/home/jabbar_y/rendu/42sh setenv MOMO foobar
jabbar_y ~/home/jabbar_y/rendu/42sh echo $MOMO
foobar
```

La commande « setenv MOMO foobar »

Elle permet d'ajouter à l'environnement, une nouvelle variable nommé « MOMO » et ayant une valeur « foobar ».

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh unsetenv DISPLAY
jabbar_y ~/home/jabbar_y/rendu/42sh echo $DISPLAY
DISPLAY: Undefined variable.
```

La commande « unsetenv DISPLAY »
Elle permet de supprimer la variable dans l'environnement nommé DISPLAY.

Ex:

```
jabbar_y ~/home/jabbar_y/rendu/42sh exit
exit
```

La commande « exit » permet de quitter l'interpréteur de commande.

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh cd include  
jabbar_y ~/home/jabbar_y/rendu/42sh/include _
```

La commande « cd include »

La commande « cd » permet de changer de dossier courant, dans ce cas elle permet d'aller dans le dossier « include »

L'option « cd - » permet de retourner dans le précédant dossier courant.

Ex:

```
jabbar_y ~/home/jabbar_y/rendu/42sh echo poney  
poney
```

La commande « echo poney »

Elle permet d'afficher « poney » suivi d'un saut à la ligne.

L'option « echo -n poney » permet de faire pareil en supprimant le saut à la ligne.

SÉPARATEURS :

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh cd include ; ls  
42sh.h
```

La commande « cd include ; ls »

Elle permet de changer de dossier courant et d'afficher celle-ci.

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh ls && tree  
42sh a.out fin include Makefile mou.sh sources tests  
.  
├── 42sh  
├── a.out  
├── fin  
├── include  
│   └── 42sh.h  
├── Makefile  
├── mou.sh  
└── sources  
    ├── alias2.c  
    ├── alias2.o  
    └── alias.c
```

La commande « ls && tree »

Cette commande exécute la commande « tree » qui affiche le contenu du dossier courant structuré en arbre si « ls » a réussi.

Ex :

```
jabbar_y ~/home/jabbar_y/rendu/42sh ls || echo fail  
42sh a.out fin include Makefile moul.sh sources tests
```

La commande « ls || echo fail »

La commande « echo » affiche son argument « fail » si la commande « ls » n'a pas réussi.

Nous pouvons assembler les séparateurs à l'infini.

3.1 FONCTIONS AVANCÉES

ALIAS :

Ex:

```
jabbar_y ~/home/jabbar_y/rendu/42sh alias l='ls'
jabbar_y ~/home/jabbar_y/rendu/42sh l
42sh a.out fin include Makefile moul.sh sources tests
```

La commande « alias ls='l' »

Elle permet de remplacer la commande « ls »,
par « l ».

Nous pouvons même éditer le fichier
« .42shrc », afin de modifier les alias.

3.1 FONCTIONS AVANCÉES

HISTORY :

Ex:

```
jabbar_y ~/home/jabbar_y/rendu/42sh history
1  ls
2  ls -l
3  emacs -nw fichier
4  echo < fichier
5  echo > fichier
6  fichier < echo
7  exit
8  < fichier
9  ls < fichier
```

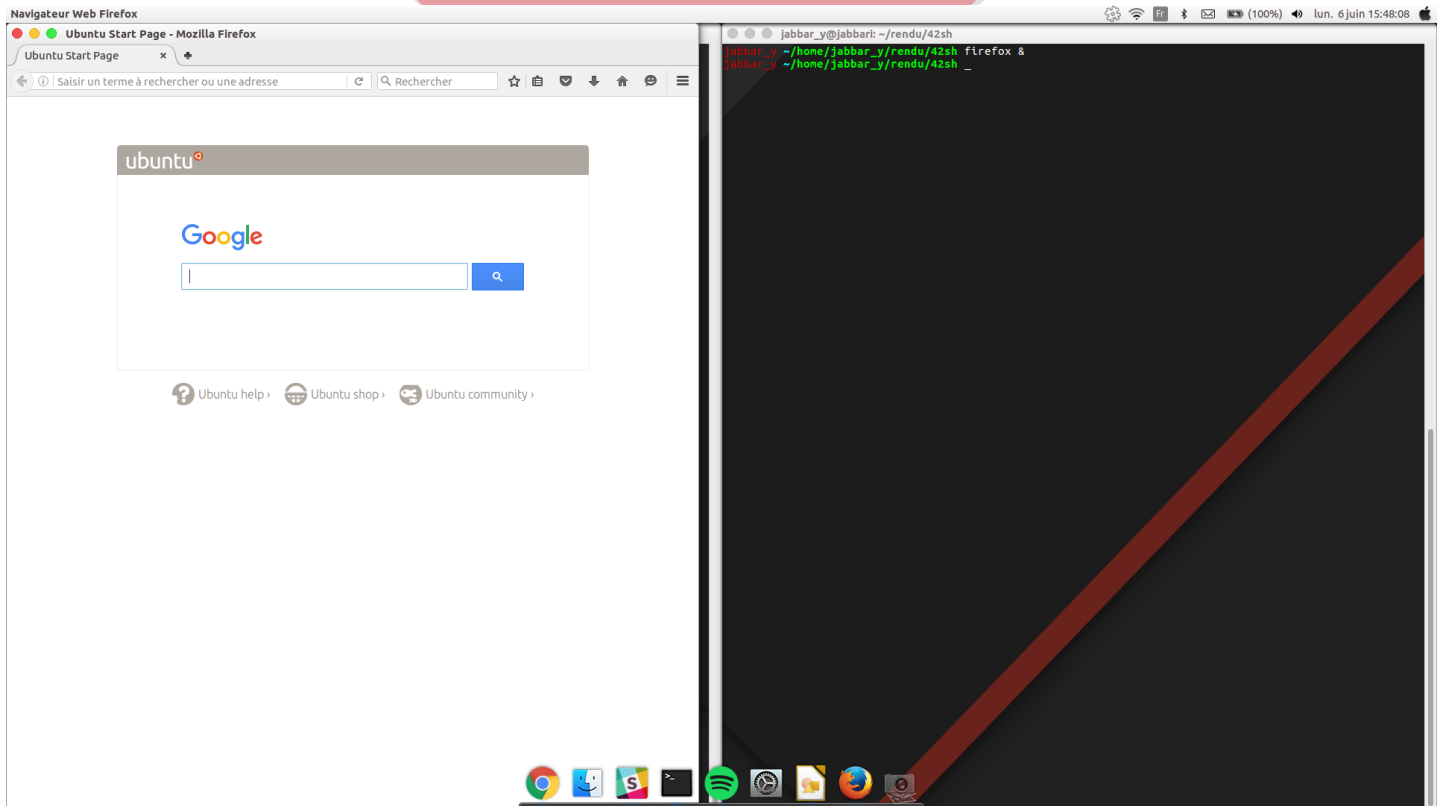
La commande « history ».

Elle permet d'afficher l'historique des commandes entrées.

L'option « -c » permet de vider l'historique.

BACKGROUND :

Ex:



La commande « firefox & »

Elle permet de lancer « firefox » avec un interpréteur de commande encore fonctionnel.