



Rapport projet d'application

IMPLEMENTATION D'UN MINI-SHELL INTERACTIF
M. DOUMBOUYA | A. BARRY | V. BUI | D. MATHIOT

Table des matières :

I- Projet initial	3
1- Travail effectué	3
2- Difficultés et raisons de changement de projet	4
II- Projet "bash-like"	6
1- Cahier des charges imposé par le professeur	6
2- Réponses au cahier des charges:	6
3- Implémentation et amélioration des commandes de l'examen :	6
a- Commande help	6
b- Commande touch	7
c- Commande add	7
d- Commande cat	7
e- Commande clean	7
f- Commande head	7
g- Commande tail	8
h- Commande cp	8
i- Commande sort	8
j- Commande find	9
k- Commande ls	9
l- Commande ls -l	10
m- Commande ls nom_fichier	10
n- Commande ls liste_nom_fichier	10
o- Commande ls -l nom_fichier	10
p- Commande ln	11
q- Commande chmod	11
r- Commande cd	11
s- Commande mkdir	11
t- Commande rmdir	11
u- Commande rm	12
v- Commande rm -r	12
w- Commande mv	13
x- Commande rmline	13
y- Commande diff	13
z- Commande grep	13
A1- Commande pwd	14
A2- Commande wc	14
A3- Commande echo	14
A4- Commande env	14
A5- Commande export	15
A6- Commande echo \$name_variable	16
A7- Commande date	16
A8- Commande du	16
A9- Commande id	16
A10- Commande sleep	16
A11- Commande cal	16
A12- Commande alias	17
A13- Commande clear/cls/ctrl+L	17

A14-	Commande history	18
A15-	Commande linecount	18
A16-	Commande jobs	18
4-	Amélioration du mini-shell (visuel, ajouter la gestion du pipe, du &&, , des flèches de redirection, la complétion de commande, rappel de commande, gestion touche ENTRER)	18
a-	Gestion du background &	18
b-	Gestion du pipe	19
c-	Gestion du &&,	19
d-	Gestion des flèches de redirection (> ; >> ; 2> ; 2>> ; <)	20
e-	Gestion de la complétion de commande	22
f-	Gestion du déplacement flèche gauche-droite sur la ligne de commande et du rappel de commande	22
g-	Gestion du ENTRER	22
5-	Limitations connues	23
6-	CONTRIBUTIONS	23

I- Projet initial :

1- Travail effectué :

Le travail initial demandé par le professeur était d'effectuer la deuxième itération d'un projet démarré il y a quelques années par des étudiants de l'**INSA**, en:

- fournissant une documentation, procédure d'installation, et une épuration de code
- mettant en place un système de fichiers (temporaire puis persistant)
- mettant en place un shell interactif avec des commandes usuelles linux

Nous avons alors commencé par appréhender le projet, en commentant et essayant de le compiler. Une première version était alors disponible avec **qemu**.



```
QEMU

Booting from Hard Disk...
loading kernel

kernel : gdt loaded
kernel : idt loaded
kernel : pic configured
kernel : tr loaded
kernel : paging enable
code size : 3907, mem size : 0
kernel : scheduler enable
Keyboard : set to fr
Bienvenue sur JuQy OS !

Shell
```

Nous avons alors créé un fichier **readme** sur la page du projet qui fournissait une procédure de mise en route de l'OS en compilant avec **docker**.

Des appels systèmes étaient implémentés, mais aucun moyen de les reconnaître. Nous avons alors passé du temps à les identifier et ainsi définir des noms pour chaque appel système comme montrés ci-dessous.

```
/* Implemented Syscalls */
#define PRINT_STRING          1
#define KILL                  2
#define SIGNAL                3
#define _UNUSED               4
#define PUTCHAR               5
#define EXEC                  6
#define GETPID                7
#define CHANGE_KEYBOARD       8
#define PRINT_SHELL           9
#define SLEEP                 10
#define NICE                   11
#define CLEAR                  12
#define PRINT_NB               13
#define RESET_MASK             14
#define MOVE_CURSOR            15
#define CHANGE_CHAR_COLOR      16
#define CHANGE_CHAR_COLOR_IN_POSITION 17
#define SHARED_MEMORY          20

/* Shared Memory Modes */
#define CREATE_SHM 0x0
#define WRITE_SHM 0x1
#define READ_SHM 0x2
#define DESTROY_SHM 0x3
#define EXIST_SHM 0x4
```

2- Difficultés et raisons de changement de projet :

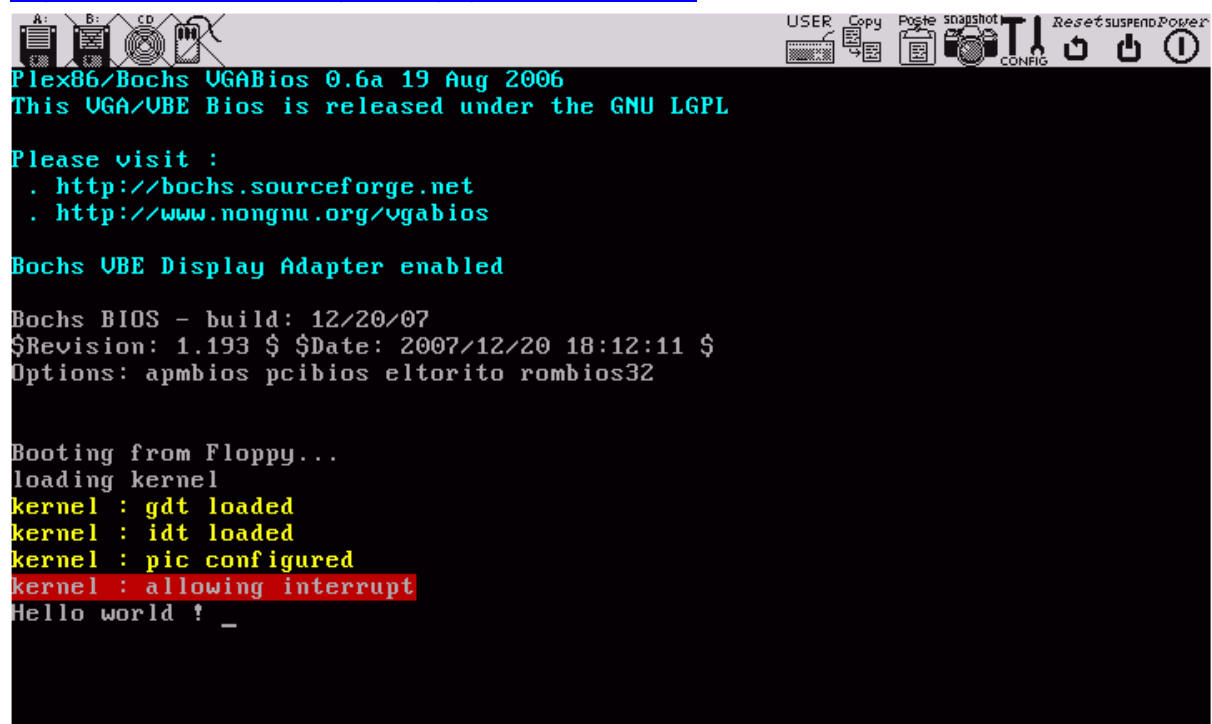
La difficulté principale que nous avons subie était l'incapacité de corriger un bug qui empêchait l'ajout de code dans les fonctions du **kernel**. Il y avait dans ce projet un problème de pagination qui faisait pointer vers des zones mémoires non allouées dès que l'on atteignait une certaine taille de code ou que l'on dépassait une page allouée.

```
Booting from Hard Disk...
loading kernel

kernel : gdt loaded
kernel : idt loaded
kernel : pic configured
kernel : tr loaded
kernel : paging enable
code size : 4023, mem size : 0
kernel : scheduler enable
Keyboard : set to fr
Bienvenue sur JuQy OS !
#PF on eip: 0x40000f46. cr2: 0x0000007f
```

Après avoir passé les 2 premiers jours à essayer de corriger ce bug, nous avons trouvé un site internet qui reprenait étrangement le code et le contenu du projet initial, mais avec certaines modifications, qui ainsi contenaient des erreurs:

<http://a.michelizza.free.fr/pmwiki.php?n=TutoOS.Kbd>



```
Plex86/Bochs UGABios 0.6a 19 Aug 2006
This VGA/VE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs VBE Display Adapter enabled

Bochs BIOS - build: 12/20/07
$Revision: 1.193 $ $Date: 2007/12/20 18:12:11 $
Options: apmbios pcibios eltorito rombios32

Booting from Floppy...
loading kernel
kernel : gdt loaded
kernel : idt loaded
kernel : pic configured
kernel : allowing interrupt
Hello world ? _
```

Après discussion avec l'équipe, nous avons souhaité en parler au professeur.

La connaissance de ce site et du code complet remet en cause le cahier des charges qui avait été demandé pour la bonne réussite du projet.

Pour cette raison, en plus de celle du problème qui nous empêchait depuis le début du projet d'ajouter du code en mémoire, il nous a été attribué un autre projet.

II- Projet “bash-like” :

1- Cahier des charges imposé par le professeur :

Le second projet est allé dans le sens d'une des **features** demandées par le professeur au projet précédent : **la mise en place d'un shell interactif**. Il nous a été alors demandé de reprendre un mini-shell donné lors de l'examen de programmation système 1, et ainsi:

- Implémenter et améliorer tous les programmes demandés par le professeur lors de l'examen
- Améliorer le mini-shell (nouvelles commandes, ajouter la gestion du pipe, du &, des flèches de redirection...)

2- Réponses au cahier des charges :

On a implémenté un terminal avec une sorte de **PS1** qui s'affiche avec les informations de l'utilisateur courant, puis **minishell** et le **dossier** dans lequel il se trouve “Inspiré de **zsh**”.

Nous avons implémenté dans ce projet une **quarantaine** de commandes linux dont certaines avec des arguments. Si vous pensez vous perdre ne vous inquiétez pas nous avons pensé à mettre un **help**. “**help** ou **cmd --help** ou **cmd -h**”.

3- Implémentation et amélioration des commandes de l'examen :

a- Commande help :

Cette commande affiche les informations nécessaires sur toutes les commandes du **mini shell** et sur une fournie en paramètre.

```
root@ns.ci.uganc.gn --minishell -- medy > help
La liste des commandes dans minishell:
add      - Ajouter des contenus dans le fichier.
alias    - Ajouter un alias au shell.
cat      - Concaténer les fichiers vers la sortie standard.
chmod    - Modifier le mode du fichier.
clean    - Nettoyer le contenu du fichier.
cp       - Copier la source vers destination.
date     - Afficher la date actuelle.
diff     - Comparer les fichiers.
echo     - Afficher une ligne de texte.
env      - Afficher l'environnement résultant.
export   - Ajouter une variable d'environnement.
find     - Rechercher le fichier dans le répertoire actuel.
grep     - Chercher des mots dans le fichier.
head     - Afficher les premières lignes du fichier sur la sortie standard.
id       - Afficher les informations d'utilisateur actuel et de groupe de l'utilisateur.
ln       - Créer un lien vers cible avec le nom.
ls       - Afficher des renseignements sur les fichiers (du répertoire actuel par défaut).
mkdir    - Créer les répertoires s'ils n'existent pas.
mv       - Déplacer le fichier vers répertoire.
pwd      - Affiche le nom du répertoire de travail courant
rm       - Supprimer (retirer le lien) les fichiers.
tail     - Afficher les dernières lignes du fichier sur la sortie standard.
touch    - Créer les nouveaux fichiers.
wc       - Afficher le nombre de sauts de ligne, de mots pour chaque fichier.
rmdir    - Supprimer les répertoires, s'ils sont vides.
history  - Afficher l'historique avec les numéros de lignes en préfixant chaque élément
cal      - Afficher un calendrier.
linecount - Afficher le contenu du fichier en préfixant chaque ligne par son numéro.
sort     - Afficher le contenu du fichier en l'ordre.
clear    - Effacer l'écran du terminal.
cls      - Effacer l'écran du terminal.
cd       - Changer le répertoire de travail du minishell.
jobs     - Afficher la liste des tâches en cours d'exécution.
alias    - Définir l'alias du commande.

Toutes les commandes prenant un fichier le prennent en premier argument
Pour utiliser le pipe il faut toujours commencer par passer le deuxième argument, le premier consiste à la sortie de la commande d'avant
root@ns.ci.uganc.gn --minishell -- medy >
```

```
root@ns.ci.uganc.gn --minishell ~ src > ls --help
ls - Afficher des renseignements sur les fichiers (du répertoire actuel par défaut).
Usage:
ls : Afficher des renseignements sur les fichiers du répertoire actuel avec le format court d'affichage
ls -l : Afficher des renseignements sur les fichiers du répertoire actuel avec le format long d'affichage
ls dir1 dir2 ... : Afficher des renseignements sur les fichiers dans les répertoire 'dir1' 'dir2'
```

b- Commande touch :

Cette commande crée un nouveau fichier. Et avec enchaînement il crée les fichiers à tour de rôle.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > touch fichier
root@ns.ci.uganc.gn --minishell ~ src >
```

c- Commande add :

Cette commande permet d'ajouter du contenu dans un fichier comme montré ci-dessous.

```
root@ns.ci.uganc.gn --minishell ~ src > add fichier
hello monsieur
comment allez-vous ?
je vais bien
<<EOF
root@ns.ci.uganc.gn --minishell ~ src >
```

d- Commande cat :

Cette commande affiche le contenu du fichier fourni en paramètre. Et si on donne plusieurs fichiers il affiche leurs contenus à tour de rôle.

```
root@ns.ci.uganc.gn --minishell ~ src > cat fichier
hello monsieur
comment allez-vous ?
je vais bien
root@ns.ci.uganc.gn --minishell ~ src >
```

e- Commande clean :

Celle-ci prend en paramètre un fichier, crée un fichier vide si il n'existe pas, ou le remet à zéro s'il existe.

```
root@ns.ci.uganc.gn --minishell ~ src > clean fichier
root@ns.ci.uganc.gn --minishell ~ src > cat fichier
root@ns.ci.uganc.gn --minishell ~ src >
```

Vous remarquerez qu'une fois avoir vidé le fichier, cat n'affiche rien.

f- Commande head :

Cette commande affiche par défaut les 10 premières lignes d'un fichier, ou les n premières lignes fournis en paramètre.


```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > head fichier
hello monsieur
comment allez-vous ?
je vais bien
root@ns.ci.uganc.gn --minishell ~ src > head fichier 2
hello monsieur
comment allez-vous ?
root@ns.ci.uganc.gn --minishell ~ src > █
```

g- Commande tail :

Cette commande affiche par défaut les 10 dernières lignes d'un fichier, ou les n dernières lignes fournis en paramètre.

```
root@ns.ci.uganc.gn --minishell ~ dossier1 > tail --help
tail - Afficher les dernières lignes du fichier sur la sortie standard.
Usage:
tail nom_fichier: Afficher les 10 dernières lignes du fichier 'nom_fichier' sur la sortie standard
tail nom_fichier nbLignes: Afficher les 'nbLignes' dernières lignes du fichier 'nom_fichier' sur la sortie standard
root@ns.ci.uganc.gn --minishell ~ dossier1 > tail fichier
hello monsieur
je vais bien
aaadjfkd
bblfjd
erer
w
xxfdj
x
12
root@ns.ci.uganc.gn --minishell ~ dossier1 > tail fichier 3
xxfdj
x
12
root@ns.ci.uganc.gn --minishell ~ dossier1 > █
```

h- Commande cp :

Cette commande effectue la copie d'un fichier.

```
root@ns.ci.uganc.gn --minishell ~ dossier1 > cp fichier fichierCopie
root@ns.ci.uganc.gn --minishell ~ dossier1 > cat fichier
hello monsieur
je vais bien
root@ns.ci.uganc.gn --minishell ~ dossier1 > cat fichierCopie
hello monsieur
je vais bien
root@ns.ci.uganc.gn --minishell ~ dossier1 > █
```

i- Commande sort :

Cette commande trie un fichier fournis en argument dans les deux ordres comme montré ci-dessous.

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ dossier1 > cat fichier
hello monsieur
je vais bien
aaadjfkd
bblfjd
erer
w
xxfdj
x
12
root@ns.ci.uganc.gn --minishell ~ dossier1 > sort fichier
Tapez 0 pour trier en ordre alphabetique inverse,
Ou 1 pour trier en ordre alphabetique direct:
Saisie > 1
12
aaadjfkd
bblfjd
erer
hello monsieur
je vais bien
w
x
xxfdj
root@ns.ci.uganc.gn --minishell ~ dossier1 > sort fichier
Tapez 0 pour trier en ordre alphabetique inverse,
Ou 1 pour trier en ordre alphabetique direct:
Saisie > 0
xxfdj
x
w
je vais bien
hello monsieur
erer
bblfjd
aaadjfkd
12
root@ns.ci.uganc.gn --minishell ~ dossier1 >

```

j- Commande find :

Cette commande nous informe si un fichier se trouve ou non dans le répertoire courant.

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ dossier1 > ls
fichierCopie      fichier
root@ns.ci.uganc.gn --minishell ~ dossier1 > find fichier
OK
root@ns.ci.uganc.gn --minishell ~ dossier1 > find fichierrrr
root@ns.ci.uganc.gn --minishell ~ dossier1 >

```

k- Commande ls :

Cette commande affiche le contenu du dossier courant en faisant une distinction entre les différents type de fichier (**exécutable, dossier, lien symbolique...**)

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls
mkdir.c      rm.c      lsfichier      du.c      id.c
date.c       cat       f2             chmod.c   echo
wc           grep      add.c          clean     mv.c
rmline       pwd.c     libc_like.o    colors.h  fich
chmod        dossier1  libc_like.h    env       Makefile
minishell.c  clean.c   date           find.c    pwd
ls           wc.c     dossier2       fichierErreur  mv
ls.c         touch.c  tail           mkdir     sleep.c
diff         grep.c   echo.c         cp         head.c
util.c       ln        sort.c         ln.c      rmline.c
cal.c        touch    export         diff.c    env.c
rmdir.c      alias.c  alias          rmdir     linecount
linecount.c  add      util.o         id         util.h
sleep        help.c   tail.c         sort       find
cal          libc_like.c  fichier      minishell help
cp.c         cat.c     head           .aliases  du
rm
root@ns.ci.uganc.gn --minishell ~ src >

```

l- Commande ls -l :

Cette commande affiche le contenu du dossier courant en ajoutant plus d'informations comme les droits sur le fichier, l'user, la taille, la date de création...

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls -l
-rwxr-xr-x 1 root root 1043 26 Jun 14:20 mkdir.c
-rwxr-xr-x 1 root root 2493 27 Jun 20:38 rm.c
-rw-r--r-- 1 root root 3016 28 Jun 00:08 du.c
-rwxr-xr-x 1 root root 953 26 Jun 14:20 id.c
-rwxr-xr-x 1 root root 834 26 Jun 14:20 date.c
-rwxr-xr-x 1 root root 17808 28 Jun 00:13 cat
-rwxr-xr-x 1 root root 941 26 Jun 14:20 chmod.c
-rwxr-xr-x 1 root root 17096 28 Jun 00:13 echo
-rwxr-xr-x 1 root root 21448 28 Jun 00:13 wc
-rwxr-xr-x 1 root root 17176 28 Jun 00:13 grep
-rwxr-xr-x 1 root root 1522 27 Jun 22:02 add.c
-rwxr-xr-x 1 root root 16984 28 Jun 00:13 clean
-rwxr-xr-x 1 root root 2030 27 Jun 12:18 mv.c
-rwxr-xr-x 1 root root 22432 28 Jun 00:13 rmline
-rw-r--r-- 1 root root 896 26 Jun 20:54 pwd.c
-rw-r--r-- 1 root root 1872 28 Jun 00:13 libc_like.o
-rw-r--r-- 1 root root 476 25 Jun 13:03 colors.h
-rwxr-xr-x 1 root root 15736 28 Jun 00:13 chmod
-rw-r--r-- 1 root root 950 27 Jun 20:38 libc_like.h
-rwxr-xr-x 1 root root 16824 28 Jun 00:13 env
-rw-r--r-- 1 root root 580 27 Jun 21:56 Makefile
-rwxr-xr-x 1 root root 13967 27 Jun 21:59 minishell.c
-rw-r--r-- 1 root root 857 27 Jun 20:38 clean.c
-rwxr-xr-x 1 root root 17304 28 Jun 00:13 date
-rwxr-xr-x 1 root root 1501 26 Jun 14:20 find.c
-rwxr-xr-x 1 root root 16800 28 Jun 00:13 pwd
-rwxr-xr-x 1 root root 24032 28 Jun 00:13 ls
-rwxr-xr-x 1 root root 1937 25 Jun 18:19 wc.c
-rwxr-xr-x 1 root root 22336 28 Jun 00:13 mv
-rwxr-xr-x 1 root root 6101 28 Jun 00:13 ls.c
-rw-r--r-- 1 root root 871 25 Jun 18:19 touch.c
-rwxr-xr-x 1 root root 21464 28 Jun 00:13 tail
-rwxr-xr-x 1 root root 16944 28 Jun 00:13 mkdir
-rw-r--r-- 1 root root 346 26 Jun 14:20 sleep.c
-rwxr-xr-x 1 root root 22152 28 Jun 00:13 diff
-rwxr-xr-x 1 root root 1370 26 Jun 14:20 grep.c

```

m- Commande ls nom_fichier :

Celle-ci affiche le nom du fichier s'il existe ou le nom du dossier et son contenu s'il existe.

```

root@ns.ci.uganc.gn --minishell ~ src > ls fichier
fichier
root@ns.ci.uganc.gn --minishell ~ src > ls dossier1/
dossier1/
fichierCopie
root@ns.ci.uganc.gn --minishell ~ src >

```

n- Commande ls liste_nom_fichier :

Notez qu'on a la possibilité sur notre shell de fournir plusieurs paramètres sur les commandes qui le prennent naturellement en charge comme la commande ls.

```

root@ns.ci.uganc.gn --minishell ~ src > ls dossier1/ dossier2/
dossier1/
fichierCopie
dossier2/
dossier      fichier3      fichier      mohamed
root@ns.ci.uganc.gn --minishell ~ src >

```

o- Commande ls -l nom_fichier :

Celle-ci affiche le nom du fichier s'il existe ou le nom du dossier et son contenu s'il existe tout en ajoutant plus d'informations.

```
root@ns.ci.uganc.gn --minishell ~ src > ls fichier
fichier
root@ns.ci.uganc.gn --minishell ~ src > ls -l fichier
-rwxr-xr-x 1 root root 62 28 Jun 01:00 fichier
root@ns.ci.uganc.gn --minishell ~ src > █
```

p- Commande ln :

Cette commande crée un lien symbolique d'un fichier fournis en argument.

```
root@ns.ci.uganc.gn --minishell ~ src > ln fichier fichiersSymbolique
root@ns.ci.uganc.gn --minishell ~ src > ls -l
lrwxrwxrwx 1 root root 7 28 Jun 00:34 fichiersSymbolique
```

q- Commande chmod :

Celle-ci change les droits d'un fichier comme montré sur l'exemple ci-dessous.

```
root@ns.ci.uganc.gn --minishell ~ src > chmod fichier 777
root@ns.ci.uganc.gn --minishell ~ src > ls -l
-rwxrwxrwx 1 root root 49 28 Jun 00:26 fichier
```

r- Commande cd :

Cette commande permet de se déplacer dans un dossier par toutes les manières (chemin absolu et relatif).

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > cd dossier1
root@ns.ci.uganc.gn --minishell ~ dossier1 > cd ..
root@ns.ci.uganc.gn --minishell ~ src > cd /home/medy
root@ns.ci.uganc.gn --minishell ~ medy > █
```

s- Commande mkdir :

Cette commande crée un dossier.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > mkdir dossier1
root@ns.ci.uganc.gn --minishell ~ src > ls
mkdir.c      rm.c         du.c         id.c         date.c
cat          chmod.c      echo         wc           grep
add.c        clean        mv.c         rmline       pwd.c
libc_like.o  colors.h    chmod        dossier1     libc_like.h
env          Makefile    minishell.c  clean.c      date
find.c       pwd         ls           wc.c         mv
ls.c         touch.c     tail         mkdir        sleep.c
diff         grep.c     echo.c      cp           head.c
util.c       ln          sort.c      ln.c         rmline.c
cal.c        touch      export      diff.c       env.c
rmdir.c      alias.c    alias       rmdir        linecount
linecount.c  add        util.o      id           util.h
fichierPro   dddd       sleep      help.c       tail.c
sort         find       cal         libc_like.c  minishell
help         cp.c       cat.c      head         aliases
fichier2     du         rm
root@ns.ci.uganc.gn --minishell ~ src > █
```

t- Commande rmdir :

Celle-ci supprime un dossier à condition qu'il soit vide.

```

root@ns.ci.uganc.gn --minishell ~ dossier1 > cd ..
root@ns.ci.uganc.gn --minishell ~ src > rmdir dossier1/
root@ns.ci.uganc.gn --minishell ~ src > ls
  mkdir.c      rm.c      du.c      id.c      date.c
  cat          chmod.c   echo      wc        grep
  add.c        clean    mv.c      rmline    pwd.c
libc_like.o   colors.h  chmod     libc_like.h  env
Makefile      minishell.c  clean.c   date      find.c
  pwd         ls        wc.c      mv        ls.c
touch.c       tail      mkdir     sleep.c   diff
grep.c        echo.c    cp        head.c    util.c
  ln          sort.c   ln.c      rmline.c  cal.c
touch         export   diff.c    env.c     rmdir.c
alias.c       alias    rmdir     linecount linecount.c
  add         util.o   id        util.h    fichierPro
  dddd        sleep   help.c    tail.c    sort
  find        cal     libc_like.c  minishell help
  cp.c        cat.c   head      .aliases  fichier2
  du          rm
root@ns.ci.uganc.gn --minishell ~ src >

```

u- Commande rm :

Cette commande supprime un fichier fourni en paramètre.

```

root@ns.ci.uganc.gn --minishell ~ src > rm fichier
root@ns.ci.uganc.gn --minishell ~ src > ls
  mkdir.c      rm.c      du.c      id.c      date.c
  cat          chmod.c   echo      wc        grep
  add.c        clean    mv.c      rmline    pwd.c
libc_like.o   colors.h  chmod     dossier   libc_like.h
  env         Makefile  minishell.c  clean.c   date
  find.c       pwd      ls        wc.c      mv
  ls.c         touch.c  tail      mkdir     sleep.c
  diff        grep.c   echo.c    cp        head.c
  util.c       ln       sort.c   ln.c      rmline.c
  cal.c        touch    export   diff.c    env.c
  rmdir.c      alias.c  alias    rmdir     linecount
linecount.c   add      util.o   id        util.h
fichierPro    dddd     sleep   help.c    tail.c
  sort        find     cal     libc_like.c  minishell
  help        cp.c    cat.c   head      .aliases
  fichier2    du      rm
root@ns.ci.uganc.gn --minishell ~ src >

```

On remarque que le ls nous montre que le fichier n'existe plus.

v- Commande rm -r :

Celle-ci quant à elle, supprime un dossier avec son contenu de façon récursive.

```

root@ns.ci.uganc.gn --minishell ~ src > mkdir dossier2
root@ns.ci.uganc.gn --minishell ~ src > cd dossier2/
root@ns.ci.uganc.gn --minishell ~ dossier2 > touch fichier2
root@ns.ci.uganc.gn --minishell ~ dossier2 > cd ..
root@ns.ci.uganc.gn --minishell ~ src > rm -r dossier2/
root@ns.ci.uganc.gn --minishell ~ src > ls
  mkdir.c      rm.c      du.c      id.c      date.c
  cat          chmod.c   echo      wc        grep
  add.c        clean    mv.c      rmline    pwd.c
libc_like.o   colors.h  chmod     libc_like.h  env
Makefile      minishell.c  clean.c   date      find.c
  pwd         ls        wc.c      mv        ls.c
touch.c       tail      mkdir     sleep.c   diff
grep.c        echo.c    cp        head.c    util.c
  ln          sort.c   ln.c      rmline.c  cal.c
touch         export   diff.c    env.c     rmdir.c
alias.c       alias    rmdir     linecount linecount.c
  add         util.o   id        util.h    fichierPro
  dddd        sleep   help.c    tail.c    sort
  find        cal     libc_like.c  minishell help
  cp.c        cat.c   head      .aliases  fichier2
  du          rm
root@ns.ci.uganc.gn --minishell ~ src >

```

Vous remarquez que le dossier *dossier2* contenant le fichier *fichier2* a été supprimé avec succès comme le montre le ls.

w- Commande mv :

Cette commande déplace un fichier vers une destination.

```
root@ns.ci.uganc.gn --minishell ~ src > mv fichier dossier1/
root@ns.ci.uganc.gn --minishell ~ src > cd dossier1/
root@ns.ci.uganc.gn --minishell ~ dossier1 > ls
fichier
root@ns.ci.uganc.gn --minishell ~ dossier1 > █
```

x- Commande rmline :

Cette commande supprime dans un fichier la ligne se trouvant à l'emplacement du numéro fourni en paramètre.

```
root@ns.ci.uganc.gn --minishell ~ src > cat fichier
hello monsieur
comment allez-vous ?
je vais bien
root@ns.ci.uganc.gn --minishell ~ src > rmline fichier 2
root@ns.ci.uganc.gn --minishell ~ src > cat fichier
hello monsieur
je vais bien
root@ns.ci.uganc.gn --minishell ~ src > █
```

y- Commande diff :

Cette nous informe si deux fichiers fournis sont différents.

```
root@ns.ci.uganc.gn --minishell ~ src >
root@ns.ci.uganc.gn --minishell ~ src > touch f1
root@ns.ci.uganc.gn --minishell ~ src > touch f2
root@ns.ci.uganc.gn --minishell ~ src > add f1
111
222
fdfd
33
777
<<EOF
root@ns.ci.uganc.gn --minishell ~ src > add f2
111
222
fdfd
3
777
<<EOF
root@ns.ci.uganc.gn --minishell ~ src > diff f1 f2

Ces fichiers ne sont pas identiques
root@ns.ci.uganc.gn --minishell ~ src > █
```

z- Commande grep :

Cette commande recherche une chaîne dans un fichier et nous affiche toutes celles qui la contiennent.

```
root@ns.ci.uganc.gn --minishell ~ src > cat fichier
hello monsieur
je vais bien
aaadjfkd
bblfjd
erer
w
xxfdj
x
12
jfk
fjd
lfdkjflkjds
jflds
fjkdflkd
fjkd
jflkd
<<EOD
klfjd
root@ns.ci.uganc.gn --minishell ~ src > grep fichier 12
12
root@ns.ci.uganc.gn --minishell ~ src > grep fichier x
xxfdj
x
root@ns.ci.uganc.gn --minishell ~ src > █
```

A1- Commande pwd :

Cette commande affiche le chemin du répertoire courant.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > pwd
/home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > █
```

A2- Commande wc :

Cette commande affiche en fonction du paramètre (-l ou -c), le nombre de ligne ou de

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ dossier1 > cat fichier
hello monsieur
je vais bien
aaadjfkd
bbfjfd
erer
w
xxfdj
x
12
root@ns.ci.uganc.gn --minishell ~ dossier1 > wc --help
wc - Afficher le nombre de sauts de ligne, de mots pour chaque fichier.
Usage:
wc -l fichier1 fichier2 ... : Afficher le nombre de sauts de ligne pour chaque fichier
wc -c fichier1 fichier2 ... : Afficher le nombre de mots pour chaque fichier
root@ns.ci.uganc.gn --minishell ~ dossier1 > wc -l fichier
fichier 9
root@ns.ci.uganc.gn --minishell ~ dossier1 > wc -c fichier
fichier 12
root@ns.ci.uganc.gn --minishell ~ dossier1 > █
```

A3- Commande echo :

Cette commande affiche ce que l'on tape sur le terminal.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > echo hello
hello
root@ns.ci.uganc.gn --minishell ~ src > █
```

A4- Commande env :

Cette commande affiche la liste des variables d'environnements.

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > env
LC_PAPER=wo_SN
LC_ADDRESS=wo_SN
LC_MONETARY=wo_SN
SHELL=/bin/bash
TERM=xterm
LC_NUMERIC=wo_SN
TREC_EVAL=/etc/trec_eval
OLDPWD=/home/medy/Bureau/ProjetA2/Bash-Like
USER=root
LC_TELEPHONE=wo_SN
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
SUDO_USER=medy
SUDO_UID=1000
USERNAME=root
MAIL=/var/mail/root
PATH=/bin:/opt/gnu/arm/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/etc/trec_eval/bin:/bin
LC_IDENTIFICATION=wo_SN
PWD=/home/medy/Bureau/ProjetA2/Bash-Like/src

```

A5- Commande export :

Celle-ci ajoute une variable d'environnement à la liste.

```

root@ns.ci.uganc.gn --minishell ~ src > export HELLO=500
root@ns.ci.uganc.gn --minishell ~ src >
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
gv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
SUDO_USER=medy
SUDO_UID=1000
USERNAME=root
MAIL=/var/mail/root
PATH=/bin:/opt/gnu/arm/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/etc/trec_eval/bin:/bin
LC_IDENTIFICATION=wo_SN
PWD=/home/medy/Bureau/ProjetA2/Bash-Like/src
EDITOR=nedit
LANG=fr_FR.UTF-8
LC_MEASUREMENT=wo_SN
SHLVL=1
SUDO_COMMAND=/bin/su
HOME=/root
LANGUAGE=fr_FR
LOGNAME=root
PREFIX=arm-eabi-
CLASSPATH=/root/.m2/repository/fr/u-ga/inf354/0.0.1-SNAPSHOT/inf354-0.0.1-SNAPSHOT.jar:
LESSOPEN=| /usr/bin/lesspipe %s
SUDO_GID=1000
DISPLAY=:0
LESSCLOSE=/usr/bin/lesspipe %s %s
LC_TIME=wo_SN
GNUARMDIR=/opt/gnu/arm
XAUTHORITY=/home/medy/.Xauthority
LC_NAME=wo_SN
_=./minishell
LINES=34
COLUMNS=100
HELLO=500
root@ns.ci.uganc.gn --minishell ~ src >

```


A6- Commande echo \$name variable :

Celle-ci affiche la valeur d'une variable d'environnement.

```
root@ns.ci.uganc.gn --minishell ~ src > echo $HELLO
500
root@ns.ci.uganc.gn --minishell ~ src > █
```

A7- Commande date :

Cette commande affiche la date et l'heure actuelles.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ dossier1 > date
Sunday 28 June 2020, 00:49:14 GMT
root@ns.ci.uganc.gn --minishell ~ dossier1 > █
```

A8- Commande du :

Cette commande affiche la liste des dossiers du répertoire courant ainsi que leur taille.

```
root@ns.ci.uganc.gn --minishell ~ src > du
4096 dossier1
4096 dossier2
root@ns.ci.uganc.gn --minishell ~ src > █
```

A9- Commande id :

Cette commande affiche les informations sur l'utilisateur actuel et son groupe.

```
root@ns.ci.uganc.gn --minishell ~ src > id
uid=0(root) gid=0(root)
root@ns.ci.uganc.gn --minishell ~ src > █
```

A10- Commande sleep :

Cette commande est utilisée pour retarder pendant un certain moment (nombre de seconde fournis en argument) le mini shell

```
root@ns.ci.uganc.gn --minishell ~ src > sleep 10
█
root@ns.ci.uganc.gn --minishell ~ src > sleep 10
root@ns.ci.uganc.gn --minishell ~ src > █
```

A11- Commande cal :

Cette commande affiche le calendrier en mettant un **highlight** sur le jour d'aujourd'hui.

```
root@ns.ci.uganc.gn --minishell ~ src > cal
-----
          JUIN 2020
-----
 L  M  M  J  V  S  D
-----
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
root@ns.ci.uganc.gn --minishell ~ src > █
```

A12- Commande alias :

Cette commande crée un alias d'une commande.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > alias cat=c
root@ns.ci.uganc.gn --minishell ~ src > c fichier
hello monsieur
je vais bien
root@ns.ci.uganc.gn --minishell ~ src > █
```

A13- Commande clear/cls/ctrl+L :

Cette commande efface le contenu de l'écran. Il est aussi possible d'appuyer sur la combinaison Ctrl+L afin d'obtenir le même résultat

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls
mkdir.c      rm.c      lsfichier    du.c      id.c
date.c       cat       f2           chmod.c   echo
wc           grep      add.c       clean     mv.c
rmline       pwd.c     libc_like.o  colors.h  fich
chmod        dossier1  libc_like.h  env       Makefile
minishell.c  clean.c   date        find.c    pwd
ls           wc.c      dossier2     fichierErreur mv
ls.c         touch.c   tail        mkdir     sleep.c
diff         grep.c    echo.c      cp        head.c
util.c       ln        sort.c      ln.c      rmline.c
cal.c        touch     export      diff.c    env.c
rmdir.c      alias.c   alias       rmdir     linecount
linecount.c  add       util.o      id        util.h
sleep        help.c    tail.c      sort      find
cal          libc_like.c fichier      minishell help
cp.c         cat.c     head        .aliases du
rm
root@ns.ci.uganc.gn --minishell ~ src > cls█
```

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > █
```

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls
mkdir.c      rm.c      lsfichier    du.c      id.c
date.c       cat       f2           chmod.c   echo
wc           grep      add.c       clean     mv.c
rmline       pwd.c     libc_like.o  colors.h  fich
chmod        dossier1  libc_like.h  env       Makefile
minishell.c  clean.c   date        find.c    pwd
ls           wc.c      dossier2     fichierErreur mv
ls.c         touch.c   tail        mkdir     sleep.c
diff         grep.c    echo.c      cp        head.c
util.c       ln        sort.c      ln.c      rmline.c
cal.c        touch     export      diff.c    env.c
rmdir.c      alias.c   alias       rmdir     linecount
linecount.c  add       util.o      id        util.h
sleep        help.c    tail.c      sort      find
cal          libc_like.c fichier      minishell help
cp.c         cat.c     head        .aliases du
rm
root@ns.ci.uganc.gn --minishell ~ src > clear█
```

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > █
```

A14- Commande history :

Cette commande affiche l'historique des précédentes commandes tapées dans le mini-shell (sans oublier la dernière commande qui est history).

```
root@ns.ci.uganc.gn --minishell ~ src > history
1: linecount fichier
2: ls
3: ls -l
4: cd dossier1
5: ls
6: cd ..
7: cat fichier
8: rm fichier
9: echo hello
10: ls -l
11: history
root@ns.ci.uganc.gn --minishell ~ src > █
```

A15- Commande linecount :

La commande linecount prend en argument un fichier, et affiche le contenu de ce fichier ainsi que l'index de chaque ligne.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > linecount fichier
1- hello monsieur
2- je vais bien
3- aaadjfkd
4- bblfjd
5- erer
6- w
7- xxfdj
8- x
9- 12
10- jfkd
11- fjfd
12- lfdkjflkjds
13- jflds
14- fjkdflkd
15- fjkd
16- jflkd
17- klfdj
root@ns.ci.uganc.gn --minishell ~ src > █
```

A16- Commande jobs :

Cette commande affiche les processus en arrière-plan, de la même manière que la commande jobs type linux.

```
root@ns.ci.uganc.gn --minishell ~ src > sleep 20 &
[1] 29029
root@ns.ci.uganc.gn --minishell ~ src > sleep 20 &
[2] 29036
root@ns.ci.uganc.gn --minishell ~ src > jobs
[1]+ 29029 En cours d'exécution sleep 20 &
[2]+ 29036 En cours d'exécution sleep 20 &
root@ns.ci.uganc.gn --minishell ~ src > █
```

4- Amélioration du mini-shell (visuel, ajouter la gestion du pipe, du &&, ||, des flèches de redirection, la complétion de commande, rappel de commande, gestion touche ENTRER) :

a- Gestion du background & :

Cette commande permet de lancer un processus en arrière-plan.

```
root@ns.ci.uganc.gn --minishell ~ src > sleep 20 &
[1] 28399
root@ns.ci.uganc.gn --minishell ~ src > █

root@ns.ci.uganc.gn --minishell ~ src > sleep 20 &
[1] 28399
root@ns.ci.uganc.gn --minishell ~ src >
[1]+  Fini      sleep 20 &
```

On remarque qu'après l'exécution du processus, un message nous informe que le processus est fini.

b- Gestion du pipe :

Le pipe nous permet de rediriger la sortie d'une commande pour ainsi l'utiliser via une autre.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls -l | wc -l
83
root@ns.ci.uganc.gn --minishell ~ src >

root@ns.ci.uganc.gn --minishell ~ src > ls | cat
mkdir.c      rm.c      lsfichier    du.c      id.c
date.c       cat       chmod.c      echo      wc
grep         add.c     clean        mv.c      rmline
pwd.c        libc_like.o colors.h     chmod     dossier1
libc_like.h  env       Makefile     minishell.c clean.c
date         find.c    pwd          ls         tmp_pipe
wc.c         dossier2  fichierErreur mv         ls.c
touch.c      tail      mkdir        sleep.c   diff
grep.c       echo.c    cp           head.c    util.c
ln           sort.c   ln.c         rmline.c  cal.c
touch        export   diff.c       rmdir.c   rmdir.c
alias.c      alias    rmdir        linecount linecount.c
add          util.o   id           util.h    sleep
help.c       fichier3 tail.c       sort      find
cal          libc_like.c fichier      minishell help
cp.c         cat.c    head        .aliases  du
rm
```

c- Gestion du &&, || :

La commande && permet de faire un 'ET' logique entre des commandes.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls && cal
mkdir.c      rm.c      lsfichier    du.c      id.c
date.c       cat       chmod.c      echo      wc
grep         add.c     clean        mv.c      rmline
pwd.c        libc_like.o colors.h     chmod     dossier1
libc_like.h  env       Makefile     minishell.c clean.c
date         find.c    pwd          ls         wc.c
dossier2     fichierErreur mv         ls.c      touch.c
tail         mkdir     sleep.c     diff       grep.c
echo.c       cp        head.c      util.c     ln
sort.c       ln.c     rmline.c    cal.c     touch
export       diff.c   rmdir.c     rmdir.c   alias.c
alias        rmdir   linecount   linecount.c add
util.o       id       util.h      sleep     help.c
fichier3     tail.c   sort        find      cal
libc_like.c  fichier  minishell   help      cp.c
cat.c        head    .aliases    du        rm

-----
      JUIN 2020
-----
 L  M  M  J  V  S  D
-----
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

root@ns.ci.uganc.gn --minishell ~ src >
```

Notez que si la première commande est invalide, on ne passera pas à la seconde.

```
root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls ffsdf && ls
ffsdf
Fichier ou dossier inexistant: No such file or directory
root@ns.ci.uganc.gn --minishell ~ src >
```

La commande || permet de faire un 'OU' logique entre des commandes.

```

root@ns.ci.uganc.gn --minishell ~ src > ls || cal
mkdir.c      rm.c      lsfichier    du.c        id.c
date.c       cat       chmod.c     echo        wc
grep         add.c     clean      mv.c        rmlne
pwd.c        libc_like.o colors.h    chmod      dossier1
libc_like.h  env      Makefile    minishell.c clean.c
date         find.c    pwd         ls          wc.c
dossier2     fichierErreur mv          ls.c       touch.c
tail         mkdir     sleep.c     diff        grep.c
echo.c       cp        head.c      util.c      ln
sort.c       ln.c      rmlne.c     cal.c       touch
export       diff.c    env.c       rmdir.c     alias.c
alias        rmdir     linecount   linecount.c add
util.o       id        util.h      sleep       help.c
fichier3     tail.c    sort        find        cal
libc_like.c  fichier  minishell   help        cp.c
cat.c        head     .aliases    du          rm

-----
      JUIN 2020
-----
 L  M  M  J  V  S  D
-----
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
root@ns.ci.uganc.gn --minishell ~ src >

```

Notez que quel que soit la commande invalide, on exécute tout de même les autres valides.

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls || fdjl
mkdir.c      rm.c      lsfichier    du.c        id.c
date.c       cat       chmod.c     echo        wc
grep         add.c     clean      mv.c        rmlne
pwd.c        libc_like.o colors.h    chmod      dossier1
libc_like.h  env      Makefile    minishell.c clean.c
date         find.c    pwd         ls          wc.c
dossier2     fichierErreur mv          ls.c       touch.c
tail         mkdir     sleep.c     diff        grep.c
echo.c       cp        head.c      util.c      ln
sort.c       ln.c      rmlne.c     cal.c       touch
export       diff.c    env.c       rmdir.c     alias.c
alias        rmdir     linecount   linecount.c add
util.o       id        util.h      sleep       help.c
fichier3     tail.c    sort        find        cal
libc_like.c  fichier  minishell   help        cp.c
cat.c        head     .aliases    du          rm

minishell : fdjl : commande introuvable
root@ns.ci.uganc.gn --minishell ~ src > fdjl || ls
minishell : fdjl : commande introuvable
mkdir.c      rm.c      lsfichier    du.c        id.c
date.c       cat       chmod.c     echo        wc
grep         add.c     clean      mv.c        rmlne
pwd.c        libc_like.o colors.h    chmod      dossier1
libc_like.h  env      Makefile    minishell.c clean.c
date         find.c    pwd         ls          wc.c
dossier2     fichierErreur mv          ls.c       touch.c
tail         mkdir     sleep.c     diff        grep.c
echo.c       cp        head.c      util.c      ln
sort.c       ln.c      rmlne.c     cal.c       touch
export       diff.c    env.c       rmdir.c     alias.c
alias        rmdir     linecount   linecount.c add
util.o       id        util.h      sleep       help.c
fichier3     tail.c    sort        find        cal
libc_like.c  fichier  minishell   help        cp.c
cat.c        head     .aliases    du          rm

root@ns.ci.uganc.gn --minishell ~ src >

```

d- Gestion des flèches de redirection (> ; >> ; 2> ; 2>> ; <) :

La commande > permet de rediriger la sortie écran dans un fichier (le fichier est vidé avant).

```

root@ns.ci.uganc.gn --minishell ~ src > ls > lsfichier
root@ns.ci.uganc.gn --minishell ~ src > cat lsfichier
mkdir.c      rm.c      lsfichier    du.c         id.c
date.c       cat       chmod.c     echo         wc
grep         add.c     clean       mv.c         rmlne
pwd.c        libc_like.o colors.h    chmod        dossier1
libc_like.h  env       Makefile    minishell.c clean.c
date         find.c    pwd         ls           wc.c
dossier2     mv        ls.c       touch.c      tail
mkdir        sleep.c   diff        grep.c       echo.c
cp           head.c    util.c      ln           sort.c
ln.c         rmlne.c  cal.c       touch        export
diff.c       env.c    rmdir.c     alias.c      alias
rmdir        linecount linecount.c add           util.o
id           util.h    sleep       help.c       fichier3
tail.c       sort      find        cal          libc_like.c
fichier      minishell help        cp.c         cat.c
head         .aliases du          rm
root@ns.ci.uganc.gn --minishell ~ src >

```

La commande >> permet de rediriger la sortie écran vers un fichier en ajoutant le résultat à la fin du fichier.

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > ls >> lsfichier
root@ns.ci.uganc.gn --minishell ~ src > cat lsfichier
mkdir.c      rm.c      lsfichier    du.c         id.c
date.c       cat       chmod.c     echo         wc
grep         add.c     clean       mv.c         rmlne
pwd.c        libc_like.o colors.h    chmod        dossier1
libc_like.h  env       Makefile    minishell.c clean.c
date         find.c    pwd         ls           wc.c
dossier2     mv        ls.c       touch.c      tail
mkdir        sleep.c   diff        grep.c       echo.c
cp           head.c    util.c      ln           sort.c
ln.c         rmlne.c  cal.c       touch        export
diff.c       env.c    rmdir.c     alias.c      alias
rmdir        linecount linecount.c add           util.o
id           util.h    sleep       help.c       fichier3
tail.c       sort      find        cal          libc_like.c
fichier      minishell help        cp.c         cat.c
head         .aliases du          rm
mkdir.c      rm.c      lsfichier    du.c         id.c
date.c       cat       chmod.c     echo         wc
grep         add.c     clean       mv.c         rmlne
pwd.c        libc_like.o colors.h    chmod        dossier1
libc_like.h  env       Makefile    minishell.c clean.c
date         find.c    pwd         ls           wc.c
dossier2     mv        ls.c       touch.c      tail
mkdir        sleep.c   diff        grep.c       echo.c
cp           head.c    util.c      ln           sort.c
ln.c         rmlne.c  cal.c       touch        export
diff.c       env.c    rmdir.c     alias.c      alias
rmdir        linecount linecount.c add           util.o
id           util.h    sleep       help.c       fichier3
tail.c       sort      find        cal          libc_like.c
fichier      minishell help        cp.c         cat.c
head         .aliases du          rm
root@ns.ci.uganc.gn --minishell ~ src >

```

La commande 2> permet de rediriger la sortie d'erreur dans un fichier (le fichier est vidé avant).

```

root@ns: /home/medy/Bureau/ProjetA2/Bash-Like/src
root@ns.ci.uganc.gn --minishell ~ src > cat 2> fichierErreur
root@ns.ci.uganc.gn --minishell ~ src > cat fichierErreur
Usage: cat filename
root@ns.ci.uganc.gn --minishell ~ src >

```

La commande 2>> permet de rediriger la sortie d'erreur à la fin d'un fichier.

```

root@ns.ci.uganc.gn --minishell ~ src > touch 2>> fichierErreur
root@ns.ci.uganc.gn --minishell ~ src > cat fichierErreur
Usage: cat filename
Usage: touch file
root@ns.ci.uganc.gn --minishell ~ src >

```

La commande < permet de lire depuis un fichier.

```
root@ns.ci.uganc.gn --minishell ~ src > cat fichier
hello monsieur
je vais bien
aaadjfkd
bbbfjd
erer
w
xxfdj
x
12
jfkjd
fjd
lfdkjflkjds
jflfs
fjkdlfkld
fjkd
jflkd
<<EOD
klfjd
root@ns.ci.uganc.gn --minishell ~ src > grep x < fichier
xxfdj
x
root@ns.ci.uganc.gn --minishell ~ src > █
```

e- Gestion de la complétion de commande :

Nous avons fait en sorte que l'utilisateur puisse appuyer sur TAB et voir la possibilité qu'il a en termes de gestion de fichier.

```
root@ns.ci.uganc.gn --minishell ~ src > cd dos
```

Au cas où il existerait plusieurs possibilités, elles le lui sont présentées.

```
root@ns.ci.uganc.gn --minishell ~ src > fm fich
fich          fichier          fichier3          fichierErreur
root@ns.ci.uganc.gn --minishell ~ src > fm fich
```

f- Gestion du déplacement flèche gauche-droite sur la ligne de commande et du rappel de commande :

Nous avons aussi rajouté la possibilité que l'utilisateur puisse se déplacer sur la commande déjà saisie et ainsi supprimer le caractère voulu et ainsi éviter de tout supprimer.

```
root@ns.ci.uganc.gn --minishell ~ dossier1 > tail fichier 3
```

Le rappel de commande a aussi été rajouté, nous le présenterons lors de la soutenance.

g- Gestion du ENTRER :

Nous avons fait en sorte que lorsque l'utilisateur tape ENTRER sur le mini shell, nous rappelons l'invite de commandes.

[illegible]

5- Limitations connues :

- On a implémenté le rappel de commandes (flèches montantes/descendantes pour parcourir l'historique) mais on a eu des soucis de dépendances car celles-ci ne sont fonctionnelles que sur les systèmes **Debian** (comme kali), et non sur **ubuntu** (autres environnements non testés).
Par souci de temps on n'a pas eu de solution de remplacement pour n'importe quel OS.
- Le formatage du **ls** sans argument "-l" au passage par un pipe montre une sorte de dysfonctionnement, cela reste tout à fait normal car la sortie du ls est formatée d'une manière à ce qu'elle soit sur **4 colonnes** et vu qu'au passage du pipe c'est une redirection de la sortie standard qui est faite vers un fichier dans lequel la prochaine commande lira, ce formatage est donc retransmis à la prochaine commande qui le traite avec le même formatage, ce qui n'est pas le cas sur les terminaux linux.
- Nous voulions aussi pouvoir enchaîner le < avec plusieurs autres commandes à la suite mais pour défaut de temps celui-là ne peut pas s'enchaîner avec d'autres commandes il s'exécute en **standalone**.

6- CONTRIBUTIONS :

Tous les membres du groupe se sont impliqués à part égal et entièrement dans la réalisation de ce projet, ce qui nous a permis de répondre parfaitement au cahier des charges avec des tâches partagées entre nous.