

Progetto di Sistemi di Elaborazione

Anno accademico 2022/2023

Realizzare il programma **myshell**, che simula il comportamento di una shell UNIX semplificata.

Il programma **myshell** deve permettere all'utente di inserire il proprio username al momento dell'avvio della shell, passandolo come argomento da linea di comando. Se l'utente non inserisce alcun username, **myshell** ne deve assegnare uno di default.

Il programma deve mostrare ciclicamente un prompt nel formato "`mariorossi@myshell#` ", dove *mariorossi* è lo username scelto dall'utente al lancio del programma (o lo username di default), e attende l'input da parte dell'utente. Quando l'utente inserisce una stringa e preme invio, il programma deve eseguire il comando corrispondente. Al termine dell'esecuzione del comando, il programma deve mostrare nuovamente il prompt.

Inoltre, solo se l'esecuzione del comando è terminata con successo, il comando stesso viene memorizzato in un file nascosto (ovvero, non visibile nell'output del comando `ls`). Tale file nascosto dovrà mantenere la lista dei comandi eseguiti da **myshell**. Ad ogni comando deve essere associato un numero di sequenza, che rappresenta l'ordine di esecuzione dei comandi stessi (il primo comando eseguito avrà numero di sequenza 0, il secondo avrà numero di sequenza 1, e così via).

I comandi che **myshell** deve supportare sono i seguenti:

- `pwd`: mostra su standard output il percorso della directory corrente. Un esempio di output prodotto dal comando è il seguente:

```
mariorossi@myshell# pwd
/home/studente/mydir
```

- `cd [dir]`: cambia la directory corrente con quella specificata dall'argomento `dir`. Se l'argomento `dir` non viene indicato, la directory corrente rimane invariata.
- `ls`: mostra su standard output il contenuto della directory corrente. L'output deve mostrare prima tutte le directory, seguite da tutti gli altri file. L'output prodotto dal comando deve avere il seguente formato:

```
mariorossi@myshell# ls
--> Directories:
- dir2
- dir1
--> Other files:
- file1
- myshell
- file2
- file3
```

- `history`: mostra su standard output il contenuto del file nascosto contenente la storia dei comandi eseguiti da **myshell**. Un esempio dell'output prodotto dal comando è il seguente:

```
mariorossi@myshell# history
0  pwd
1  ls
2  cd dir1
3  pwd
4  ls
```

- `exit`: termina il programma, mostrando su standard output un messaggio di saluto per l'utente. Un esempio dell'output prodotto dal comando è il seguente:

```
mariorossi@myshell# exit
Goodbye, mariorossi!
```

Dove possibile, i comandi devono essere implementati come *programmi esterni* al programma **myshell**.

Se l'utente preme CTRL+C il programma, anziché terminare, deve mostrare su standard output un messaggio che informa l'utente che per terminare il programma deve digitare il comando `exit`.

Mettere in atto delle politiche di gestione delle eventuali situazioni di fallimento delle diverse system call e funzioni di libreria invocate dal programma. Fare in modo che sia possibile la redirectione dei soli messaggi di errore eventualmente prodotti dal programma **myshell**.

Un esempio di esecuzione del programma è il seguente:

```
studente@debian-SdE:~/mydir$ ./myshell mariorossi
Welcome to myshell, mariorossi!
mariorossi@myshell# pwd
/home/studente/mydir
mariorossi@myshell# ls
--> Directories:
- dir2
- dir1
--> Other files:
- file1
- myshell
- file2
- file3
mariorossi@myshell# cd dir1
mariorossi@myshell# pwd
/home/studente/mydir/dir1
mariorossi@myshell#
mariorossi@myshell#
mariorossi@myshell# ls
--> Directories:
--> Other files:
- fileA
- fileB
mariorossi@myshell# cd dirA
Cannot change directory
mariorossi@myshell# history
0      pwd
1      ls
2      cd dir1
3      pwd
4      ls
mariorossi@myshell# ^C
To close myshell, type 'exit'
mariorossi@myshell# exit
Goodbye, mariorossi!
studente@debian-SdE:~/mydir$
```