## 6. Simple shell 0.1.1

Simple shell 0.1 +

- Write your own `getline` function
- Use a buffer to read many chars at once and call the least possible the `read` system call
- You will need to use `static` variables
- You are not allowed to use `getline`

You don't have to:

- be able to move the cursor

**Repo:**

- GitHub repository: `simple_shell`

## 7. Simple shell 0.2.1

Simple shell 0.2 +

- You are not allowed to use `strtok`

**Repo:**

- GitHub repository: `simple_shell`

## 8. Simple shell 0.4.1

Simple shell 0.4 +

- handle arguments for the built-in `exit`
- Usage: `exit status`, where `status` is an integer used to exit the shell

```
julien@ubuntu:~/shell$ ./shell_0.4.1
$ exit 98
julien@ubuntu:~/shell$ echo $?
98
julien@ubuntu:~/shell$
```

**Repo:**

- GitHub repository: `simple_shell`

## 9. setenv, unsetenv

Simple shell 1.0 +

Implement the `setenv` and `unsetenv` builtin commands

- `setenv`
  - Initialize a new environment variable, or modify an existing one
  - Command syntax: `setenv VARIABLE VALUE`
  - Should print something on stderr on failure
- `unsetenv`
  - Remove an environment variable
  - Command syntax: `unsetenv VARIABLE`
  - Should print something on stderr on failure

**Repo:**

- GitHub repository: `simple_shell`

## 10. cd   #advanced

Simple shell 1.0 +

Implement the builtin command `cd` :

- Changes the current directory of the process.
- Command syntax: `cd [DIRECTORY]`
- If no argument is given to `cd` the command must be interpreted like `cd $HOME`
- You have to handle the command `cd -`
- You have to update the environment variable `PWD` when you change directory

`man chdir` , `man getcwd`

**Repo:**

- GitHub repository: `simple_shell`

## 11. ;                                                                    #advanced

Simple shell 1.0 +

- Handle the commands separator `;`

```
alex@~$ ls /var ; ls /var
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$ ls /hbtn ; ls /var
ls: cannot access /hbtn: No such file or directory
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$ ls /var ; ls /hbtn
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
ls: cannot access /hbtn: No such file or directory
alex@~$ ls /var ; ls /hbtn ; ls /var ; ls /var
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
ls: cannot access /hbtn: No such file or directory
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$
```

**Repo:**

- GitHub repository: `simple_shell`

## 12. && and ||

Simple shell 1.0 +

- Handle the `&&` and `||` shell logical operators

```
alex@~$ ls /var && ls /var
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$ ls /hbtn && ls /var
ls: cannot access /hbtn: No such file or directory
alex@~$ ls /var && ls /var && ls /var && ls /hbtn
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
ls: cannot access /hbtn: No such file or directory
alex@~$ ls /var && ls /var && ls /var && ls /hbtn && ls /hbtn
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
ls: cannot access /hbtn: No such file or directory
alex@~$
alex@~$ ls /var || ls /var
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$ ls /hbtn || ls /var
ls: cannot access /hbtn: No such file or directory
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$ ls /hbtn || ls /hbtn || ls /hbtn || ls /var
ls: cannot access /hbtn: No such file or directory
ls: cannot access /hbtn: No such file or directory
ls: cannot access /hbtn: No such file or directory
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$ ls /hbtn || ls /hbtn || ls /hbtn || ls /var || ls /var
ls: cannot access /hbtn: No such file or directory
ls: cannot access /hbtn: No such file or directory
ls: cannot access /hbtn: No such file or directory
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  spool  tmp
alex@~$
```

## 13. alias  #advanced

Simple shell 1.0 +

- Implement the `alias` builtin command
- Usage: `alias [name[='value'] ...]`
    - `alias` : Prints a list of all aliases, one per line, in the form `name='value'`
    - `alias name [name2 ...]` : Prints the aliases `name` , `name2` , etc 1 per line, in the form `name='value'`
    - `alias name='value' [...]` : Defines an alias for each `name` whose `value` is given. If `name` is already an alias, replaces its value with `value`

**Repo:**

- GitHub repository: `simple_shell`

## 14. Variables

Simple shell 1.0 +

- Handle variables replacement
- Handle the $? variable
- Handle the $$ variable

```
julien@ubuntu:~/shell$ ./hsh
$ ls /var
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  snap  spool  tmp
$ echo $?
0
$ echo $$
5104
$ echo $PATH
/home/julien/bin:/home/julien/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gam
es:/snap/bin
$ exit
julien@ubuntu:~/shell$
```

**Repo:**

- GitHub repository: simple_shell

## 15. Comments

Simple shell 1.0 +

- Handle comments ( # )

```
julien@ubuntu:~/shell$ sh
$ echo $$ # ls -la
5114
$ exit
julien@ubuntu:~/shell$
```

**Repo:**

- GitHub repository: simple_shell

## 16. File as input

#advanced

Simple shell 1.0 +

- Usage: `simple_shell [filename]`
- Your shell can take a file as a command line argument
- The file contains all the commands that your shell should run before exiting
- The file should contain one command per line
- In this mode, the shell should not print a prompt and should not read from `stdin`

**Repo:**

- GitHub repository: `simple_shell`