

shell

A C language implemented shell, Author:gaotaku, Date:04,Oct,2016.

(To keep validation rule of my friend's project, code will be released after 8:00AM CST 05,Oct,2016)

Easy tasks

- constant loop
test by *gaotaku*

```
| (just use it)
```

- simple command
test by *gaotaku*:

```
| ls  
| ls -l
```

- **more complex commands(problematic)**
test by *gaotaku*:

```
| sudo apt-get update
```

- pipe
test by *gaotaku*:

```
| ls -l | grep shell | wc -l
```

- file I/O redirection
test by *gaotaku*:

```
| echo 123 > 1.txt  
| echo 465 >> 1.txt  
| cat < 1.txt
```

- more commands
test by *gaotaku*:

```
| vi 2.txt
```

- more complex commands with pipes

test by *gaotaku*:

```
ls | xargs grep .txt
```

- clean exit, no memory leak(**partly ignored**)

test by *gaotaku*:

```
valgrind ./shell
ls -l | grep ctb | wc -l
```

ctrl+d ignored by *gaotaku*:

this will call for a keyboard catching function, I don't know how to do it.

for implementation, just go to head of function loop(), and insert keyboard catching codes and return 0 (be careful with potential memory leak).

Medium tasks

- pipe composed with file I/O redirection

test by *gaotaku*:

```
cat < 1.txt | sort -R > 2.txt
```

- display working directory

test by *gaotaku*

```
pwd
```

- wait for command > and |

test by *gaotaku*:

```
ls -l |
wc -l |
wc -l
```

- handle quotes

test by *gaotaku*:

```
echo "123 \" ' \""
```

- catch errors

test by *gaotaku*:

cat <

Hard tasks

- change directory
test by *gaotaku*:

cd ..

- **ctrl + c(ignored)**
ignored by *gaotaku*:
this will call for a keyboard catching function, I don't know how to do it.
for implementation, just go to module `//run it with fork()//`, and insert keyboard catching and child terminating codes to father part of `fork()`.

Attention

- in real bash command '|', '>', '<' and etc. should be regard as single commands, but I am too lazy to implement this, so, each word of all commands should be splitted by `r"\s+"`.