Machine Problem 1 : Building a shell

Team name : Manner Maketh Man

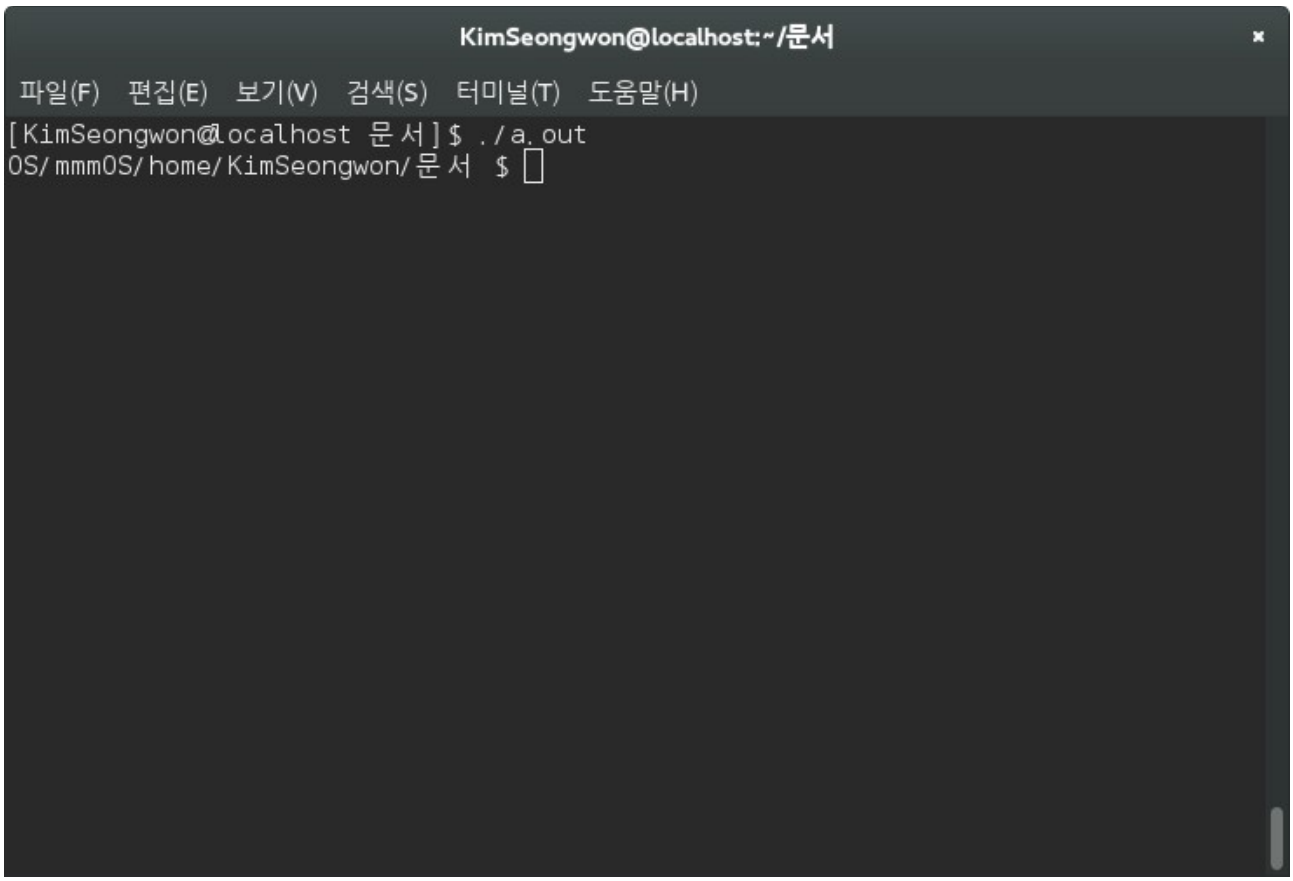
Member : kim han sung(20112944), kim seong won(20112618)

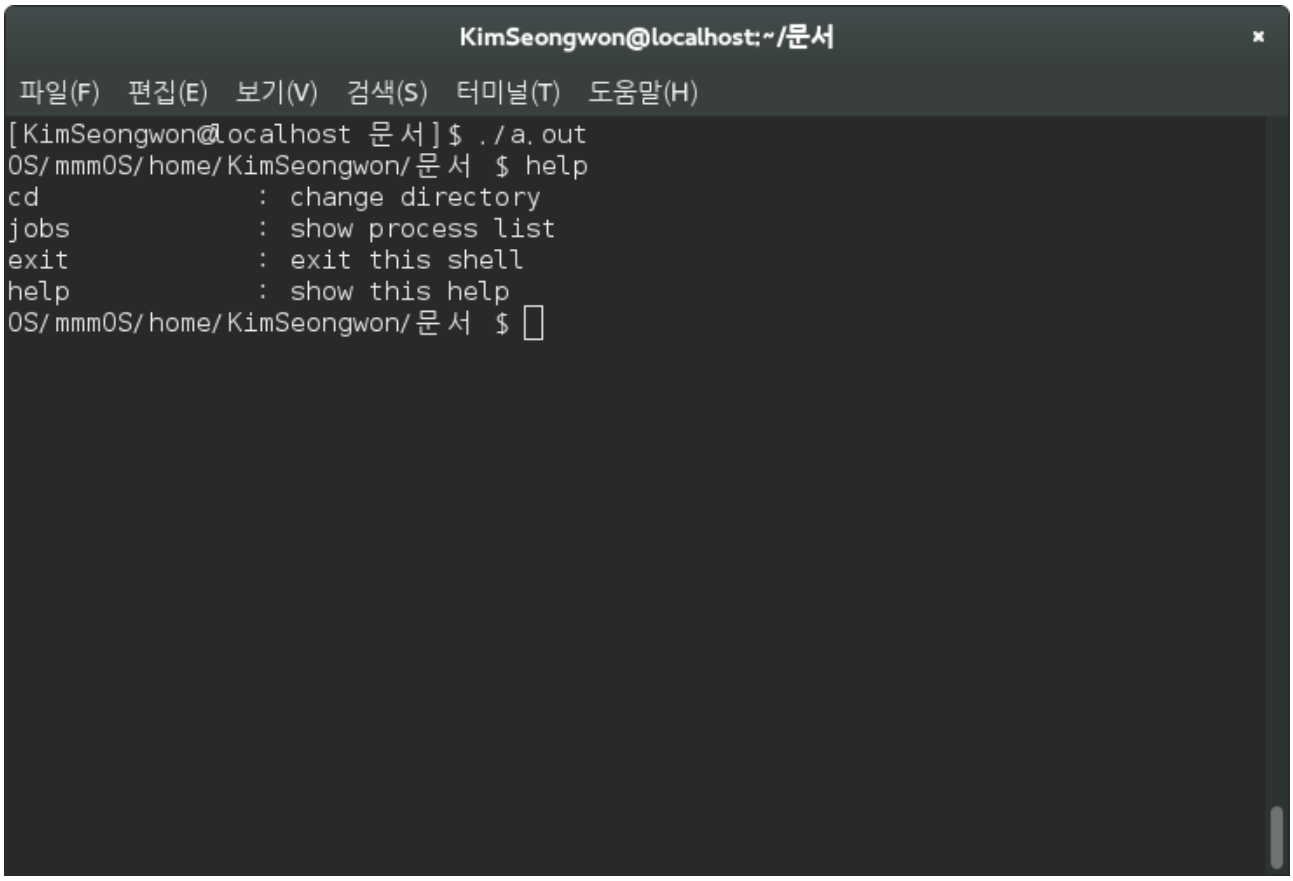
1. Discuss OS responsibilities for process management, in other words, for process creation, process deletion, process suspension, and inter-process communication, describe what they are and how the Unix supports them with associated system calls.
- Is responsible for the management of the OS, Processor Scheduling. Because you should divide works to CPU for executing logically continuous . Of course, running indefinitely rather refers to a process that must be stopped while the other process running. This way is called a context switching, in process position it is suddenly way . Therefore, the OS must be stored somewhere in the processes that were done to conceal the sudden whiff of this and proceed back and retrieve information that was saved after going to another job. This transition takes place to store important information and paging occurs when a system call and proceed to other processes. When a process is performed by the CPU interrupts except the OS are not able to interrupt the process system call yourself stuck for a moment the OS calls the OS internal function.

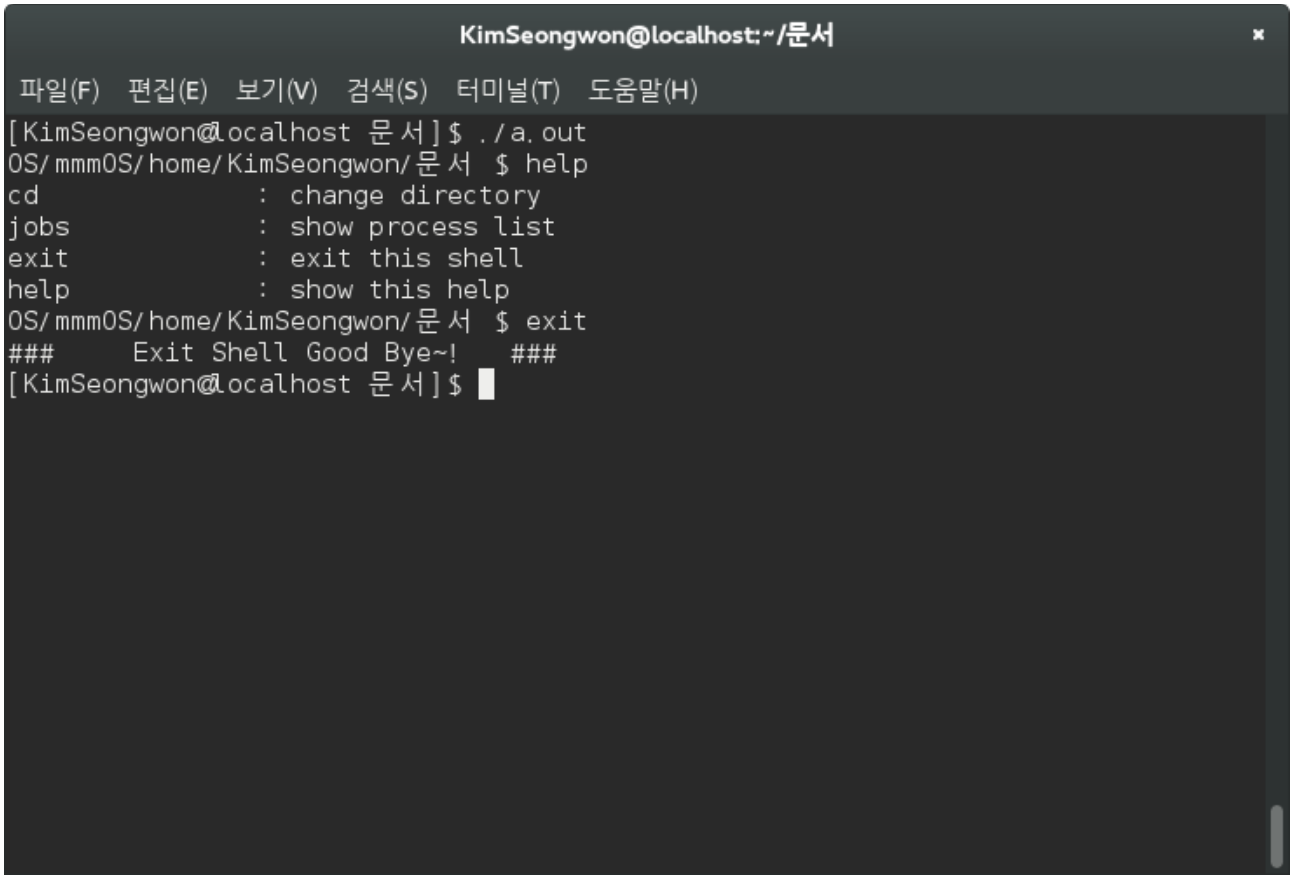
2. Discuss what internal commands are, how they are different external commands, and how such difference is reflected in your implementation.
- Both are cmd in file,but internal cmd is recalled from memory when the first OS to start working, but external cmd load on demand has continued to file. For internal cmd like cd or ls is called often and light loading on memory is good, but external cmd is not called and heavy loading on memory is not good. Therefore, cd, jobs, exit are implemented in the program go up to memory waitting for other commands, other commands on the memory is executed by execvp functions searching commands on file when necessery.

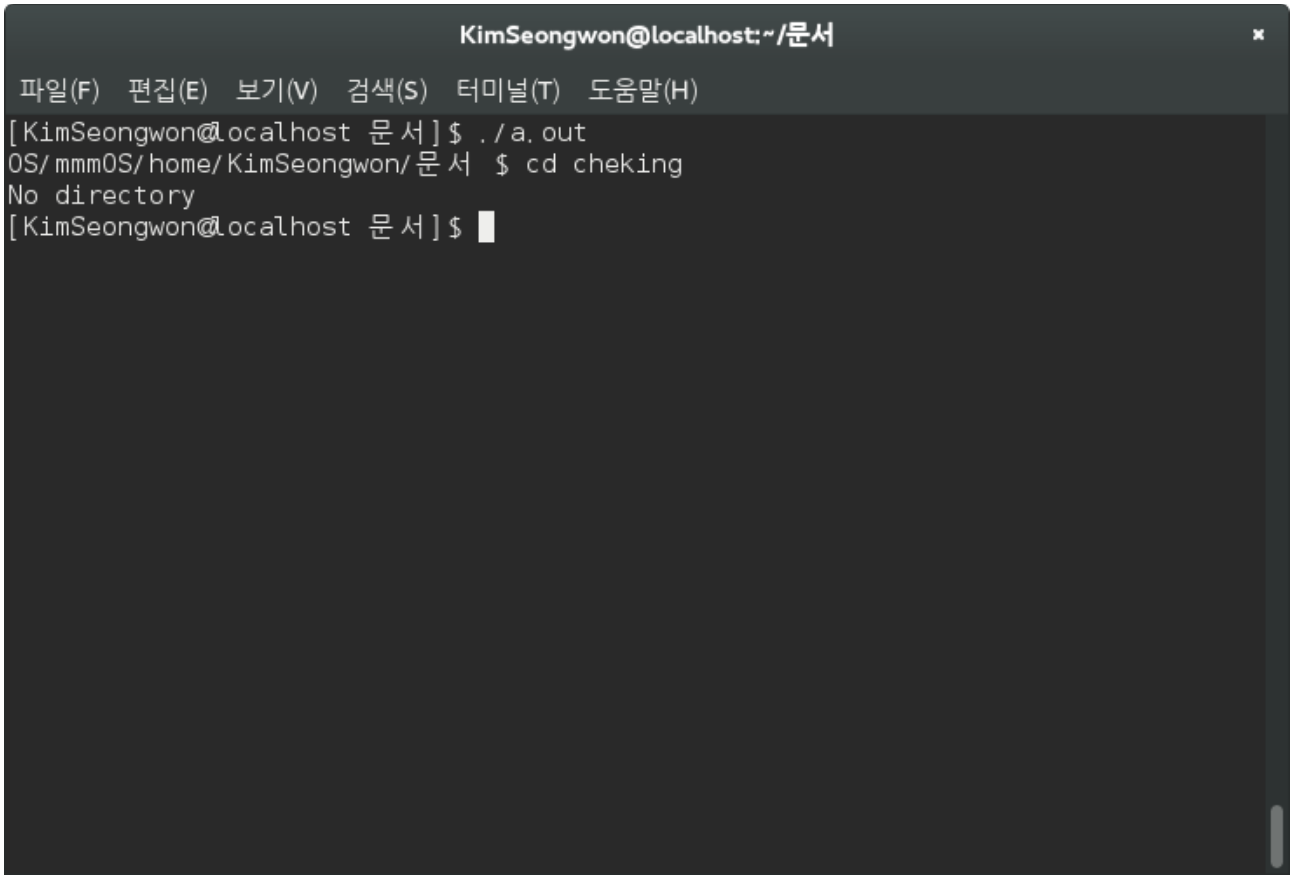
3.Discuss the difference between the foreground and background processes and how such difference is reflected in your implementation.
- While Foreground cannot interupt until other process end, background can interupt while other process processing. If '&' is in input value in implemented code, it goes to background. In case of foreground wait until process end by calling waitpid, background can process without waitting

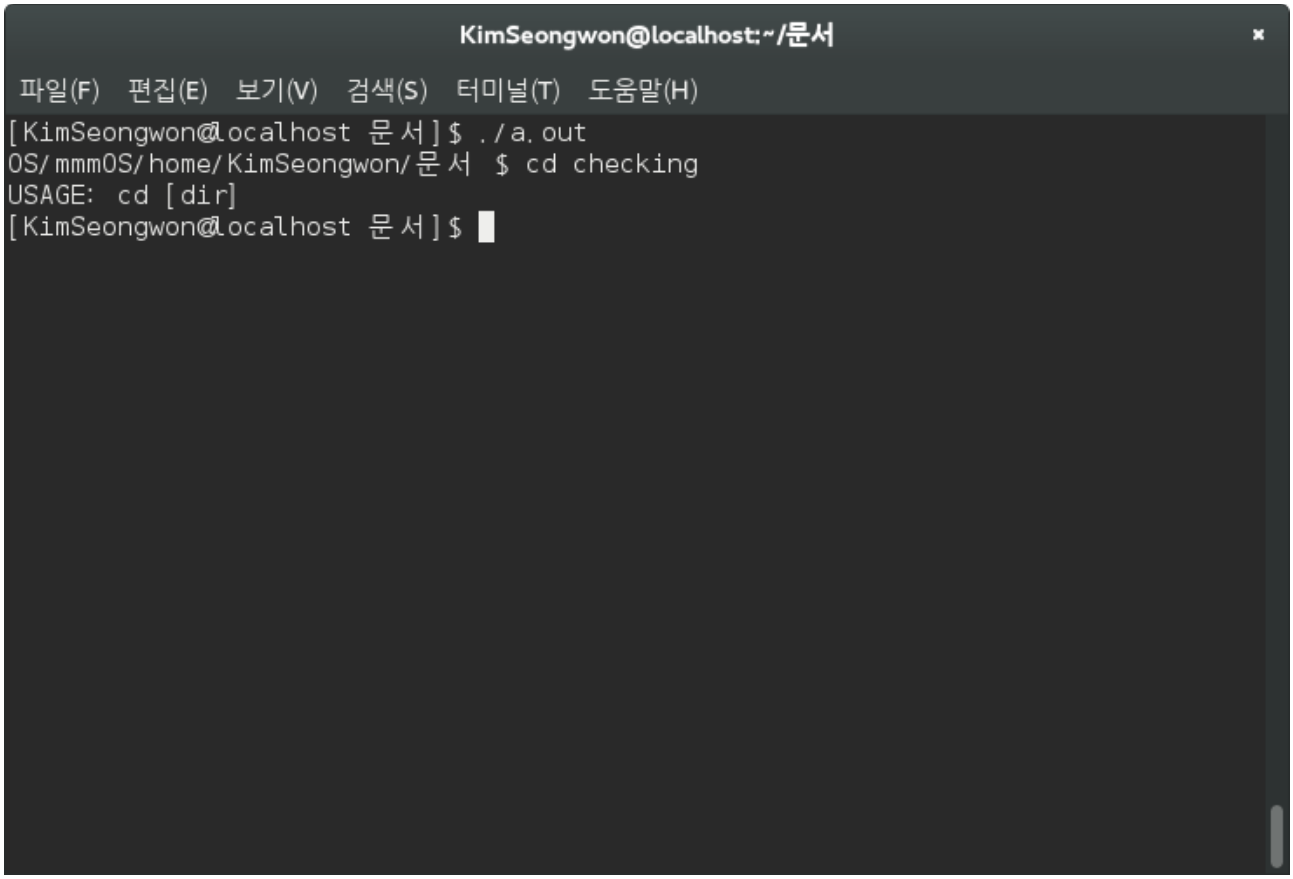
0) your working environment and assignment details to each team member.
We use LINUX(Ubuntu and Fedora) as OS. We used vim and gcc. Kim Hansung made cd, jobs main, run,and KimSeongwon made exit, help, linesplit and command sturucture.
1)implementation details about each requirement,
At first print "OS/mmmOS%s $ ", %s is directory that program stored. When shell get input value, check empty or not. If empty, back to main. Input value is one of cd, exit, help, jobs, go to right function. Case of cd, check home or whether right directory or not. If home print 'Home', existing directory make new path to directory, not existing print error. When input value is exit quit process. If input value is help, print name of commands and their describe. Jobs show all process. When client use any program it make new process as child. When it fail print error message and quit.
2)the screenshots of the results after executing them.

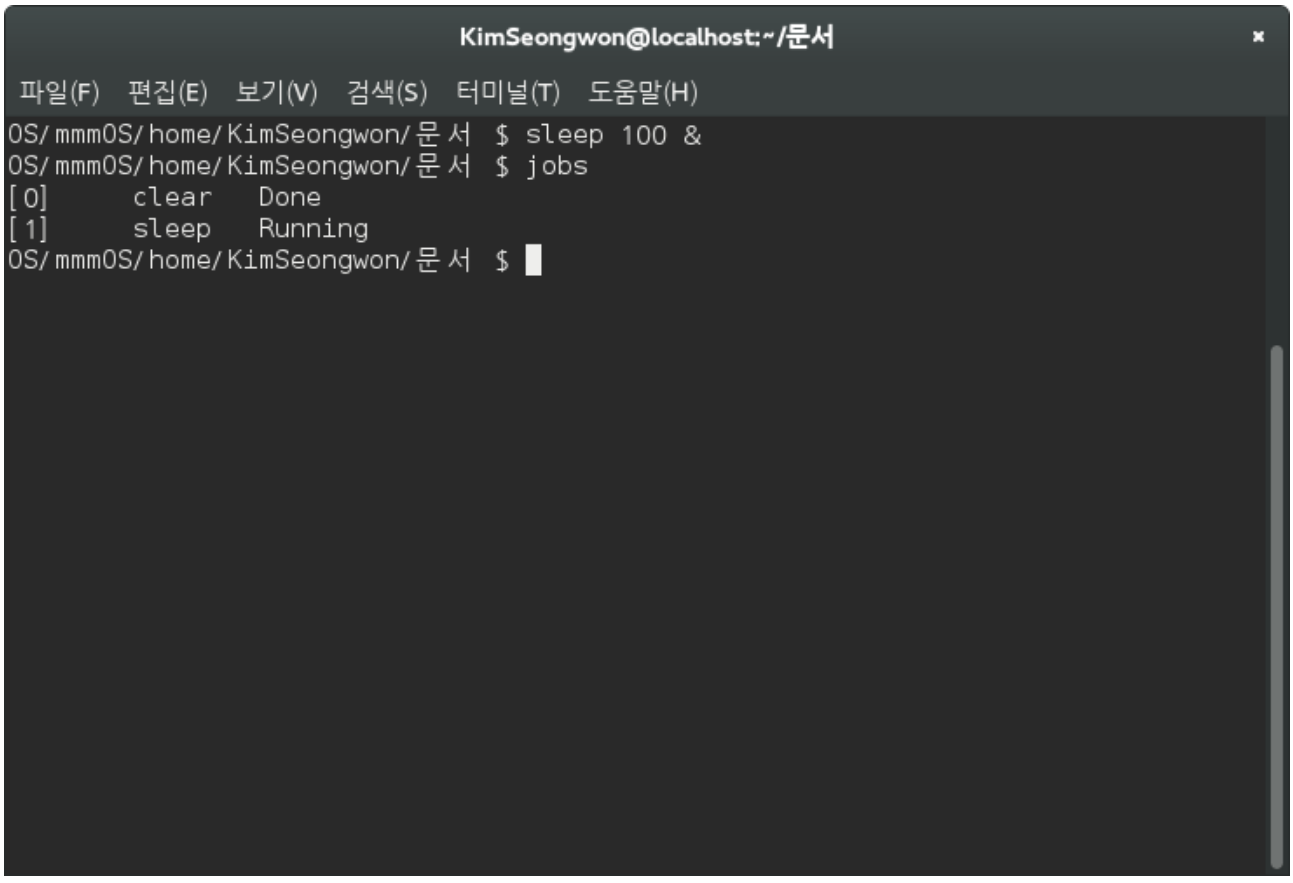

KimSeongwon@localhost:~/문서
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[KimSeongwon@localhost 문서]$ ./a.out
OS/mmmOS/home/KimSeongwon/문서 $


KimSeongwon@localhost:~/문서
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[KimSeongwon@localhost 문서]$ ./a.out
OS/mmmOS/home/KimSeongwon/문서 $ help
cd : change directory
jobs : show process list
exit : exit this shell
help : show this help
OS/mmmOS/home/KimSeongwon/문서 $


KimSeongwon@localhost:~/문서
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[KimSeongwon@localhost 문서]$ ./a.out
OS/mmmOS/home/KimSeongwon/문서 $ help
cd : change directory
jobs : show process list
exit : exit this shell
help : show this help
OS/mmmOS/home/KimSeongwon/문서 $ exit
Exit Shell Good Bye~!
[KimSeongwon@localhost 문서]$


KimSeongwon@localhost:~/문서
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[KimSeongwon@localhost 문서]$ ./a.out
OS/mmmOS/home/KimSeongwon/문서 $ cd cheking
No directory
[KimSeongwon@localhost 문서]$


KimSeongwon@localhost:~/문서
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[KimSeongwon@localhost 문서]$ ./a.out
OS/mmmOS/home/KimSeongwon/문서 $ cd checking
USAGE: cd [dir]
[KimSeongwon@localhost 문서]$


KimSeongwon@localhost:~/문서
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
OS/mmmOS/home/KimSeongwon/문서 $ sleep 100 &
OS/mmmOS/home/KimSeongwon/문서 $ jobs
[0] clear Done
[1] sleep Running
OS/mmmOS/home/KimSeongwon/문서 $