

학번 : 2016024739 이름 : 김기환

1. 기타 함수 설명

A. `Int main(int argc, char **argv)`

- i. shell처럼 돌아가는 프로세스의 역할을 한다.
- ii. User가 원하는 명령어를 입력하면 그 입력을 받아서 처리하는 역할을 함.

B. `Int parseline(const char *cmdline, char **argv)`

- i. 입력된 `cmdline`을 `parse`하는 함수로 만약 `bg`로 돌리라는 명령어가 입력되면 `true`를 `fg`로 돌리라는 명령어가 입력되면 `false`를 리턴한다.
- ii. 또한 해당 명령어에 `&`가 있는지 없는지도 체크하여 `bg`로 돌아갈지 말지를 알 수 있다.

C. `Void do_bgfg(char **argv)`

- i. 매개인자 `argv`에 있는 커맨드를 `bg`혹은 `fg`로 돌리는 함수.

D. `Pid_t fgpid(struct job_t *jobs)`

- i. 현재 foreground job의 프로세스 id를 리턴한다. 만약 foreground에서 돌아가는 job이 없으면 0을 반환한다.

E. `Void listjobs(struct job_t *jobs)`

- i. 현재 돌아가고 있는 job들의 리스트를 `stdout`으로 출력해주는 함수

F. `Int addjob(struct job_t *jobs, pid_t pid, int state, char* cmdline)`

- i. Job list에 job을 추가 해주는 함수

G. `Int deletejob(struct job_t *jobs, pid_t pid)`

- i. Job list에서 pid에 해당하는 job 제

2. 제작한 함수

A. Int builtin_cmd(char ** argv)

- i. Builtin_cmd안에서 처리해야되는 경우는 4가지가 있습니다.
 - 1. Quit이 들어온 경우 exit()이라는 시스템 콜을 이용하여 종료
 - 2. Jobs가 들어온 경우 background에서 실행되는 jobs들을 나열한다
 - 3. Bg가 들어온 경우 해당 job에 해당되는 프로세스를 bg로 변경
 - 4. Fg가 들어온 경우 해당 job에 해당되는 프로세스를 fg로 변경

B. Void waitfg(pid_t pid)

- i. 구현해야 하는 것
 - 1. foreground에서 매개인자의 pid가 끝났는지 안끝났는지를 체크한다.
- ii. 구현방법
 - 1. While을 계속 돌면서 해당 fgpid함수를 통해 해당 pid가 계속 체크하게 됩니다.

C. Void eval(char *cmdline)

- i. Str[MAXARGS]
 - 1. Argv에 해당되는 string 배열을 저장하는 변수
- ii. Bg
 - 1. 명령어가 background로 돌아갈 경우를 체크하는 변수
- iii. Pid
 - 1. 자식프로세스인지 부모 프로세스인지 구분하기 위한 변수
- iv. 구현 방법
 - 1. 먼저, bg = parseline(cmdline, str)을 통해 해당 명령어가 background로 돌아가야 되는지 아닌지를 체크하고 str에 parsing된 결과를 보내준다

2. `Builtin_cmd(str)`을 통해 해당 명령어가 `quit`, `bg`, `fg`를 처리하게 합니다.
3. `Fork()` 시스템 콜을 통해 프로세스를 소환하는데 이때 자식프로세스에는 `setpgid`라는 함수를 호출하여 자식프로세스가 새로운 그룹프로세스id에 속하도록 합니다.
4. `Execve`를 통해 해당 명령어의 파일을 실행하게 합니다.
5. `Fork()` 시스템 콜을 통해 소환된 부모 프로세스는 `joblist`를 관리하고 만약 `fg`로 돌아가야되는 명령어 이면 `waitfg`를 통해 해당 `job`이 끝날 때 까지 기다리게 됩니다.