

## Lab6 Assignment 보고서

20180038

박형규

### eval function

먼저 `parseline` 함수를 통해 `cmdline`을 파싱하고, bg인지 아닌지를 판단해주는 값을 return 해와서 `is_bg` variable에 저장한다. 그리고 `builtin_cmd` 함수를 실행시켜서 `builtin_cmd`가 아닌 경우, 즉 리턴값이 0인 경우 `mask`로 `SIGCHLD`를 block하도록 한다. 그 이후 `fork()`를 통하여 자식 프로세스를 만든다. `fork() < 0`일 경우에는 `unix_error`을 통하여 에러메세지를 출력하고, 자식 프로세스일 경우에는 `mask`의 block을 풀고 `setpgid` 함수를 통해 `pgid` 값을 현재 프로세스의 `pid` 값으로 지정하고 현재 프로세스를 그룹리더로 지정한다. 그리고 부모 프로세스일 경우 먼저 foreground 상태일 때는 FG state의 job을 추가해주고 child가 terminate될 때까지 기다린다. background 상태일 경우에는 BG state의 job을 추가해준다.

### 'builtin\_cmd' function

`builtin command`로는 `argv[0]`으로 `quit`, `jobs`, `bg`, `fg` command가 들어오는 것이 해당된다. 먼저 `quit`이 들어올 경우 `exit(0)`을 이용해서 종료해준다. `jobs`가 들어오면 `listjobs` 함수를 통해 `joblist`를 print해준다. `bg`나 `fg`가 들어올 경우 `do_bgfg` 함수를 통해 현재 state가 `fg`라면 `bg`로 바꿔주고, 현재 state가 `bg`라면 `fg`로 바꿔준다.

### 'do\_bgfg' function

먼저 `argv[1]`이 NULL일 경우엔 에러메세지를 출력하고 return한다.

`argv[0]`이 %일 경우에는 `jobid`가 입력된 경우이다. 따라서 `jobid`를 `atoi` 함수를 이용하여 받아오고, job을 `getjobid` 함수를 이용하여 받아온다. 이때, job이 NULL일 경우에는 에러메세지를 출력하고 return해준다. job이 NULL이 아닐 경우에는 `pid`를 받아온 job의 `pid`로 저장해준다.

`argv[1][0]`을 `isdigit` 함수의 인자로 넘겨주어 0이 아닌 값이 return될 경우, 즉 `argv[1][0]`이 숫자 문자가 아닐 경우는 `pid`가 입력된 경우이다. 따라서 `pid`를 `atoi` 함수를 이용하여 받아오고, job을 `getjobpid` 함수를 이용하여 받아온다. 이때, job이 NULL일 경우에는 에러메세지를 출력하고 return해준다.

다른 모든 경우에는 `jid`와 `pid` 모두 해당되지 않으므로 에러메세지를 출력하고 return해준다. 그 이후에는 `argv[0]`이 `fg`일 경우에는 background job을 foreground로 바꿔야 하므로 job의 state를 FG로 바꾸고 `kill` 함수를 통해 job의 `pid`와 같은 `id`를 가진 모든 그룹 멤버에게 `SIGCONT` signal을 준다. 그리고 `waitfg` 함수를 실행시킨다. `argv[0]`이 `bg`일 경우에는 foreground job을 background로 바꿔야 하므로 job의 state를 BG로 바꾸고 job의 정보를 출력시키고 `kill` 함수를 통해 job의 `pid`와 같은 `id`를 가진 모든 그룹 멤버에게 `SIGCONT` signal을 준다.

### 'waitfg' function

while문의 조건에 true를 주어서 무한루프를 돌게 한다. 그리고 `fgpid` 함수를 통해 jobs에 해당하는 `pid`를 가져오고 이것이 인자로 넘어온 `pid`와 동일하지 않은 경우에 process `pid`가 더

이상 foreground의 process와 같지 않은 경우이므로 if(verbose) 조건문을 통해 break시켜 주어 루프에서 나오도록 한다. 나머지 경우엔 sleep함수를 통하여 기다리게 한다.

‘sigchld\_handler’ function

while문의 조건에서 waitpid함수를 호출하여 zombie children을 reap하는 과정을 시작한다. 먼저 WIFEXITED의 반환값이 0이 아니라면, 즉 정상적으로 종료되었다면 deletejob함수를 이용하여 job을 삭제한다. 다음으로 WIFSIGNALED의 return값이 0이 아니라면, 즉 자식 프로세스가 어떤 signal에 의하여 종료 되었다면 해당 문구를 print하고 job을 삭제한다. 마지막으로 WIFSTOPPED의 return값이 0이 아니라면, 즉 자식 프로세스가 현재 정지된 상태라면 pid에 해당하는 job의 state를 ST로 설정하고 해당 문구를 출력해준다.

‘sigint\_handler’, ‘sigstp\_handler’ function

fgpid함수를 통해 foreground에 있는 pid를 가져온다. 그리고 가져온 pid가 valid하다면, 즉 pid가 0이 아니라면 kill함수를 이용하여 pid를 가진 group에 sig를 보내준다.