**Lab 9 shell lab**

**20200445박은하**

이번 랩은 signal함수를 이해하고, shell을 작성하는 것이었다.

**eval>**

eval함수에서는 cmdline을 parseline함수로 처리하여 background에서 일어나는 명령인지 bg라는 변수에 기록한다. builtin command인지 builtin\_cmd라는 함수를 통해 확인하고 아닐 시 SIGCHLD를 block하고 fork한다. 만약 자식 프로세스인 경우 setpgid(0,0)을 호출하여 새로운 프로세스 그룹의 리더가 자신이 된다. 다시 sigprocmask를 이용해 blocking을 해제하고 자식 프로세스를 실행시킨다. 이때 해당하는 command가 없을 경우 오류 메시지를 출력하고 종료한다. 만약 built in command가 아니면서 foreground에서 일어나는 job일 경우 job의 state를 FG로 하여 addjob을 호출하고 SIGCHLD 를 unblock한다. waitfg함수를 호출하여 Process가 foreground의 job이 아닐때까지 기다린다.

만약 background job일경우 job의 state를 BG로 하여 addjob을 호출하고 SIGCHLD를 unblock한다. 이때는 waitfg함수를 호출할 필요가 없다. 왜냐하면 background는 실행시켜놓고 다음 command를 받을 수 있기 떄문이다.

**builtin\_cmd>**

만약 argv[0]이 quit인경우 프로그램을 종료하고, fg이거나 bg인 경우 do\_bgfg를 호출한다. 만약 jobs인경우 listjobs(jobs)를 호출한다. Fg,bg,jobs의 경우 함수 호출하고 return값은 1이다. 그 외의 경우는 builtin command가 아니므로 0을 return한다.

**do\_bgfg >**

Error handling에 유의해야 하는 함수이다.

만약 bg나 fg 이후에 나오는 argument가 존재하지 않거나 적절한 형식을 갖추지 않으면 error message를 출력하고 return한다.

만약 jid값을 argument로 주면 getjobjid를 통해 job\_t를 얻는다. 만약 jid에 해당하는 job이 없을 경우 error message를 출력하고 return한다. 만약 pid값을 argument로 주면 유사하게 getjobpid를 통해 job\_t의 pointer를 얻는다. 이 또한 pid에 해당하는 job이 없을 경우 error message를 출력하고 return한다. 만약 argv[0]이 bg일 경우 state를 BG로 설정하고, job의 상태를 출력한다. 또한 해당하는 process group에 SIGCONT를 전달한다. fg일경우 state를 FG로 설정하고, 해당하는 process group에 kill함수를 통해 SIGCONT를 전달한다.

**wait\_fg>**

Foreground의 job의 경우 현재 실행되고 있는 foreground job이 종료되어야 다음 command를 받을 수 있다. Wait\_fg함수는 pid값을 인자로 받아 현재 foreground의 pid와 같은지 비교하고 아닐 경우 return한다. 이 때 비교하는 시간 간격을 두기 위해 sleep함수를 호출한다.

**sigchld\_handler>**

Waitpid의 option을 WNOHANG|WUNTRACE로 두어 대기 집합 모든 자식 프로세스들이 정지하였거나 종료하였다면 리턴값 0으로 즉시 리턴되거나 자식들 중 한 개의 PID의 동일한 값으로 리턴한다. 따라서 return된 pid값으로 해당하는 job을 getjobpid함수로 찾는다.

종료되거나 정지된 자식의 상태를 WIFEXITED WIFSIGNALED WIFSTOPPED로 분류했는데 WIFEXITED의 경우 deletejob을 호출했다. WIFSIGNALED의 경우 몇번째 signal에 의해 종료되었는지 출력하고 job을 delete했다.

만약 WIFSTOPPED의 경우 job의 state를 ST로 바꾸고 몇번째 signal에 의해 stop 되었는지 출력했다.

**sigint\_handler>**

현재 foreground가 아닐경우 pid값이 0이므로 그냥 return하게 하고 foreground일 경우 해당되는 pid의 process group에 sig를 보낸다.

**sigstp\_handler>**

현재 foreground가 아닐경우 pid값이 0이므로 그냥 return하게 하고 foreground일 경우 해당되는 pid의 process group에 sig를 보낸다.