2018 시스템 프로그래밍

- Lab 07 -

제출일자	2018.11.19
분 반	02
이 름	진승언
학 번	201404377

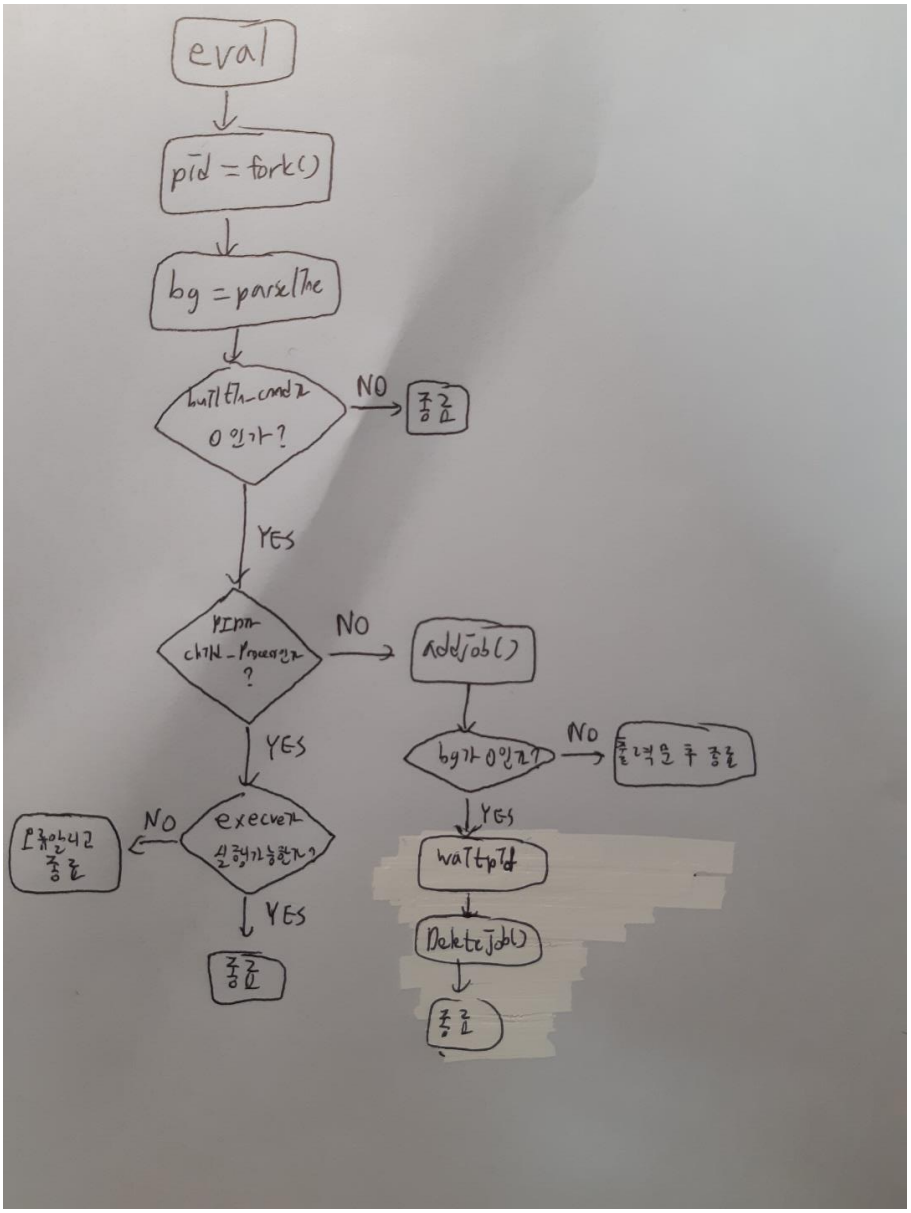
실습 5,6 [화면 캡처]

```
c201404377@2018-sp:~/shlab-handout$ ./sdriver -V -t 05 -s ./tshref
Running trace05.txt...
Success: The test and reference outputs for trace05.txt matched!
Test output:
#
# trace05.txt - Run a background job.
#
tsh> ./myspin1 &
(1) (4368) ./myspin1 &
tsh> quit

Reference output:
#
# trace05.txt - Run a background job.
#
tsh> ./myspin1 &
(1) (4376) ./myspin1 &
tsh> quit

c201404377@2018-sp:~/shlab-handout$ ./sdriver -V -t 06 -s ./tshref
Running trace06.txt...
Success: The test and reference outputs for trace06.txt matched!
Test output:
#
# trace06.txt - Run a foreground job and a background job.
#
tsh> ./myspin1 &
(1) (4697) ./myspin1 &
tsh> ./myspin2 1

Reference output:
#
# trace06.txt - Run a foreground job and a background job.
#
tsh> ./myspin1 &
(1) (4706) ./myspin1 &
tsh> ./myspin2 1
```


eval
pid = fork()
bg = parseline
builtin_cmd가 0인가?
NO
종료
YES
builtin_cmd가 Process인가?
NO
addjob()
YES
bg가 0인가?
NO
출력을 후 종료
YES
execve가 실행가능한가?
NO
오류메시지 종료
YES
종료
waitpid()
Deletejob()
종료

실습 5,6 [과정 설명]

```
170 void eval(char *cmdline)
171 {
172     char *argv[MAXARGS]; //comand 저장
173     pid_t pid = fork(); //process ID
174     int bg;
175     //명령어를 parseline을 통해 분리
176     bg = parseline(cmdline, argv);
177
178
179     if(!builtin_cmd(argv)){
180         if(pid == 0){//Child Process 인 경우, execve()수행
181             if((execve(argv[0], argv, environ) < 0 )){
182                 printf("%s : Command not found\n", argv);
183                 exit(0);
184             }
185         }
186
187         addjob(jobs, pid, (bg == 1? BG:FG),cmdline);
188
189
190         if(!bg){
191             //부모프로세서가 자식프로세서의 종료를 대기 후 처리
192             int status;
193             waitpid(pid, &status, 0 );
194             deletejob(jobs, pid);
195             }
196         }else{
197             printf("(%d) (%d) %s", pid2jid(pid), pid, cmdline);
198         }
199
200     }
201     return;
202 }
```

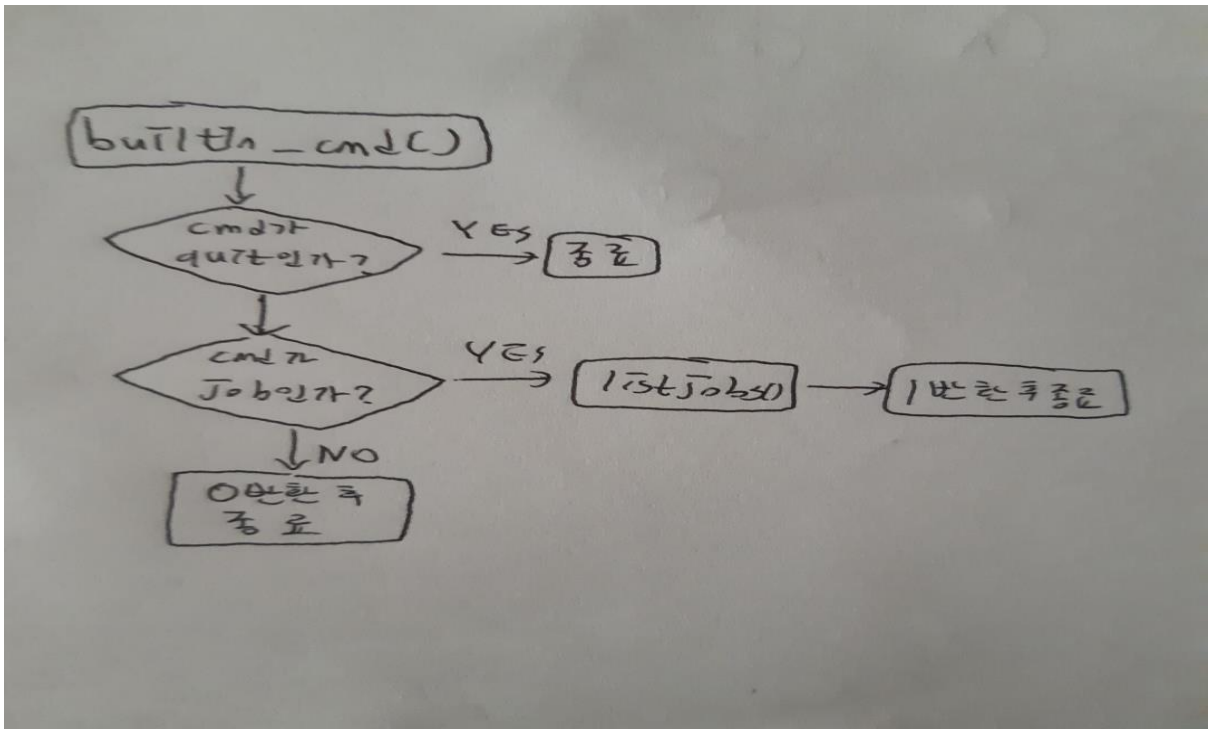
trace05는 background 작업 형태로 프로그램을 실행시키는 것이고 trace06은 동시에 forefround 작업 형태와 background 작업 형태로 프로그램을 실행시키는 것이다. 먼저 명령어를 파싱한 결과를 bg변수에 저장한다. 그 후 bulitin_cmd가 0을 반환하면 계속 실행하고 0을 반환하지 않으면 종료한다. 0이 맞다면 계속하여 자식 프로세스인지를 확인하고 아니라면 bg에 따라 실행결과가 또 나뉜다. bg는 parseline 함수결과 값을 받은 변수인데 parseline함수는 마지막 인자가 &이면 1을 아니면 0을 반환한다. 마지막 인자가 &이면 background 작업을 의미한다. 아니라면 0을 반환한다. 먼저 addjob를 통해 작업 리스트에 작업을 추가한다. foreground인 경우는 프로세스 하나만 실행 될 수 있으므로 작업관리가 필수적이므로 background의 경우 addjob함수를 구현했다.

bg값이 &가 있어 1이 되면 위와 같이 출력문을 남기고 0이라면 waitpid로 자식 프로세스가 종료되길 기다린 후, deletejob으로 해당 자식 프로세스의 구조체를 delete하고 종료시킨다.

실습 7 [화면 캡처]

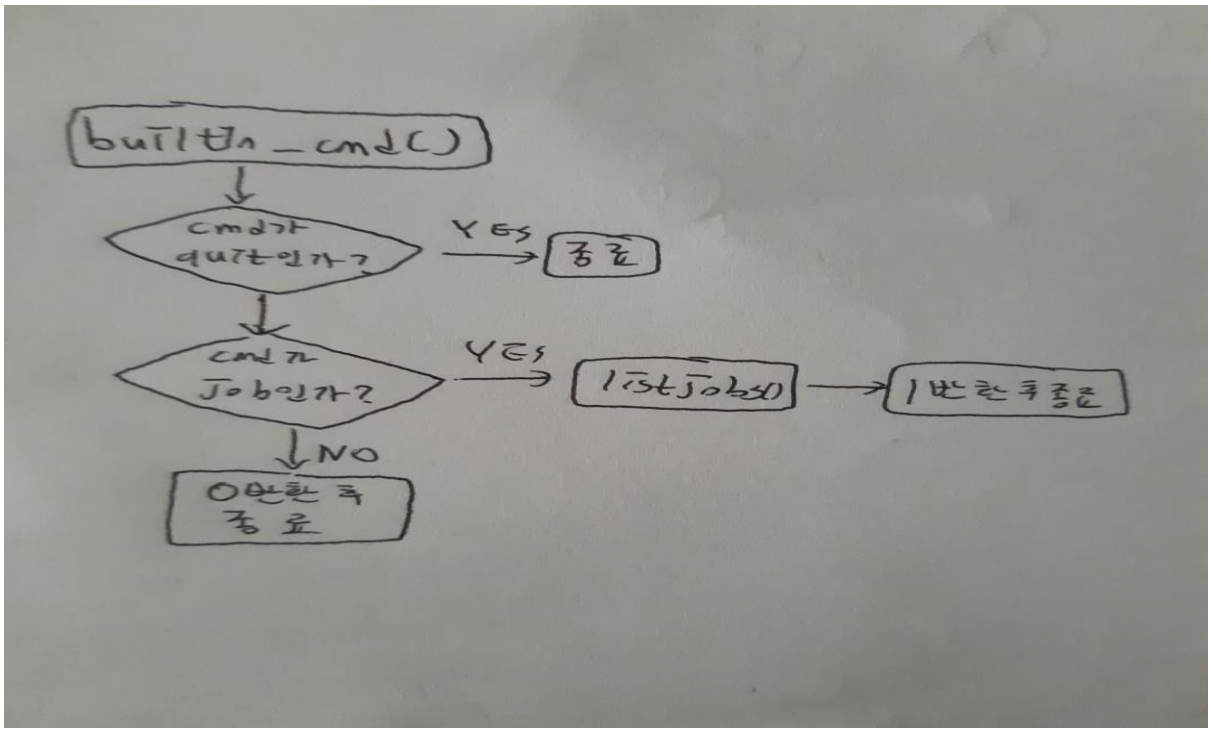
```
c2014043777@2018-sp:~/shlab-handout$ ./sdriver -V -t 07 -s ./tshref
Running trace07.txt...
Success: The test and reference outputs for trace07.txt matched!
Test output:
#
# trace07.txt - Use the jobs builtin command.
#
tsh> ./myspin1 10 &
(1) (24824) ./myspin1 10 &
tsh> ./myspin2 10 &
(2) (24826) ./myspin2 10 &
tsh> jobs
(1) (24824) Running ./myspin1 10 &
(2) (24826) Running ./myspin2 10 &

Reference output:
#
# trace07.txt - Use the jobs builtin command.
#
tsh> ./myspin1 10 &
(1) (24834) ./myspin1 10 &
tsh> ./myspin2 10 &
(2) (24836) ./myspin2 10 &
tsh> jobs
(1) (24834) Running ./myspin1 10 &
(2) (24836) Running ./myspin2 10 &
```



실습 7 [과정 설명]

```
204 int builtin_cmd(char **argv)
205 {
206     char *cmd = argv[0];
207
208     if(!strcmp(cmd, "quit")){ /* quit command */
209         exit(0);
210     }
211     else if(!strcmp(cmd, "jobs")){
212         listjobs(jobs, NULL);
213         return 1;
214     }
215
216
217     return 0; /* not a builtin command */
218 }
```



trace07은 두 개의 background 작업을 실행한 후, built-in 명령어 'jobs'을 실행한다. 해당 명령어를 입력 받으면, 현재 실행되고 있는 작업의 리스트를 출력해주는 것이다.

builtin_cmd함수에다가 cmd가 quit이면 종료를 하고 jobs인 경우는 listjobs()을 실행시킨다. 이 함수는 작업리스트를 출력하기 위해 쓰인다. 그 후 1을 반환해준다. 만약 cmd가 jobs와 quit둘 다아닌경우는 0을반환 후 종료한다.