

## Lab 6 (Shell Lab) Report

학번 : 20220312

이름 : 박준혁

### 명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

### 1. 프로그램 개요

- 간단한 shell을 구현해 본다. 작성한 기능은 eval, builtin\_cmd, do\_bgfg, waitfg, sigchild\_handler, sigint\_handler, sigtstp\_handler 이다.

### 2. 프로그램의 구조 및 알고리즘

| eval  |   |
|---|---|
| <pre>void eval(char *cmdline) {     //use argv to start process     char *argv[MAXARGS];     //background check     int bg;     //signal mask setting     sigset_t mask;     sigemptyset(&amp;mask);     sigaddset(&amp;mask, SIGCHLD);     //process id     pid_t pid;      //parse cmd line     bg = parseline(cmdline,argv);     //empty line check     if(argv[0]==NULL){         return;     } }</pre> | <p>우선 인자들을 받을 argv, background job인지 판단할 bg, 시그널 블록을 위한 mask, 프로세스 id를 위한 pid를 정의</p> <p>이후 parseline을 호출해 argv에 값을 받아온다.</p> |

|   |   |
|---|---|
| <pre>//execute if(!builtin_cmd(argv)){ //if it was built-in     //blocking     sigprocmask(SIG_BLOCK, &amp;mask, NULL);     //fork child     pid = fork();      //error     if(pid&lt;0){         unix_error("forking error");     } }</pre>  | <p>이제 builtin cmd를 호출한다.<br/>만약 builtin이었다면 그대로 실행<br/>아니면 아래에서 child를 fork한다.</p>  |
| <pre>//child if(pid==0){     //unblock     sigprocmask(SIG_UNBLOCK, &amp;mask, NULL);     setpgid(0,0);     //execute     if(execvp(argv[0], argv) &lt; 0) {         printf("%s: Command not found\n", argv[0]);         exit(0);     } } //parent if(pid&gt;0){     if(bg){         addjob(jobs, pid, BG, cmdline);     }     else{         addjob(jobs, pid, FG, cmdline);     }     //unblock     sigprocmask(SIG_UNBLOCK, &amp;mask, NULL);      if(bg){         printf("[%d] (%d) %s", pid2jid(pid), pid, cmdline);     }     else{         waitfg(pid);     } }</pre> | <p>child인 경우에는<br/>블록을 해제해 주고,<br/>execvp를 호출해 명령을 실행</p> <p>parent인 경우에는<br/>background foreground를 분류해<br/>호출한 뒤</p> <p>블록을 해제해 주고</p> <p>bg면 실행 결과를 알려주고<br/>fg면 wait을 호출한다.</p> |
| builtin_cmd   |   |
| <pre>int builtin_cmd(char **argv) {     //quit, jobs, bg or fg     //quit     if(!strcmp(argv[0], "quit")){         exit(0);     }     //jobs     if(!strcmp(argv[0], "jobs")){         listjobs(jobs);         return 1;     }     //bg, fg     if(!strcmp(argv[0], "bg")    !strcmp(argv[0], "fg")){         do_bgfg(argv);         return 1;     }     return 0;    /* not a builtin command */ }</pre>  | <p><b>builtin_cmd</b></p> <p>명령어를 strcmp로 분석해 실행<br/>quit이면 바로 exit</p> <p>jobs면 listjobs 호출</p> <p>bg, fg면 do_bgfg를 호출해<br/>job을 실행한다.</p>   |

| do_bgfg   |   |
|---|---|
| <pre> void do_bgfg(char **argv) {     struct job_t* job;     int jid;     int pid;      //no argument     if(argv[1] == NULL) {         printf("%s command requires PID or %%jobid argument\n", argv[0]);         return;     }     //pid     if(isdigit(argv[1][0])){         pid=atoi(argv[1]);         job=getjobpid(jobs, pid);         if(job==NULL){             printf("(%d): No such process\n", pid);             return;         }     }     //jid     else if (argv[1][0] == '%') {         jid = atoi(&amp;argv[1][1]);         job=getjobjid(jobs, jid);         if (job==NULL){             printf("%s: No such job\n", argv[1]);             return;         }         else{             pid = job-&gt;pid;         }     }     //nor pid jid     else{         printf("%s: argument must be a PID or %%jobid\n", argv[0]);         return;     } } </pre> | <p><b>do_bgfg</b></p> <p>argument가 없을 시 에러 발생</p> <p>pid로 호출한 경우(숫자)<br/>프로세스로 처리<br/>만약 없는 pid면 에러 발생</p> <p>jid로 호출한 경우(%)<br/>job으로 처리한다<br/>만약 없는 jid면 에러 발생</p> <p>둘다 아니면 에러를 발생시킨다.</p> |
| <pre> //killing if(kill(-pid, SIGCONT) &lt; 0) {     unix_error("kill error at do_bgfg"); }  //waiting for foreground if(!strcmp("fg", argv[0])) {     job-&gt;state = FG;     waitfg(job-&gt;pid); }  //print for bg else if(!strcmp("bg", argv[0])){     printf("[%d] (%d) %s",            job-&gt;jid, job-&gt;pid, job-&gt;cmdline);     job-&gt;state = BG; }  //error else{     printf("error in do_bgbf: %s\n", argv[0]); }  return; </pre>  | <p>이제 멈춘 프로세스를 다시 실행한다</p> <p>fg의 경우 wait을 호출<br/>state를 fg로 설정한다.</p> <p>bg의 경우 출력 해주고<br/>state를 bg로 설정한다.</p> <p>둘다 아닌 경우 에러를 발생</p>   |

| waitfg  |  |
|---|--|
| <pre> void waitfg(pid_t pid) {     while(pid == fgpid(jobs)){         sleep(1);     }     return; } </pre>  | fg가 끝날 때까지 기다린다  |
| sigchld_hadler  |  |
| <pre> void sigchld_handler(int sig) {     int status;     pid_t pid;     //waiting     //WNOHANG: return immediately when no children     //WUNTRACED: return when process is STOPPED     while((pid = waitpid(fgpid(jobs), &amp;status,          WNOHANG   WUNTRACED)) &gt; 0){         if(WIFEXITED(status)){             deletejob(jobs, pid);         }         else if(WIFSTOPPED(status)){             printf("Job [%d] (%d) stopped by signal %d\n",                 pid2jid(pid), pid, WSTOPSIG(status));             getjobpid(jobs, pid)-&gt;state = ST;         }         else if(WIFSIGNALED(status)){             printf("Job [%d] (%d) terminated by signal %d\n",                 pid2jid(pid), pid, WTERMSIG(status));             deletejob(jobs, pid);         }     }     return; } </pre> | <p>child의 상태에 따라 달라진다.<br/>exit로 끝났다면 job을 제거해주고</p> <p>stop되었다면 출력 하고<br/>state를 stop으로 바꿔주고</p> <p>시그널을 받았다면 출력하고<br/>job을 제거해 준다.</p> |
| sigint_handler, sigstp_handler  |  |
| <pre> void sigint_handler(int sig) {     pid_t pid = fgpid(jobs);      //check any jobs     if (pid&gt;0) {         //kill foreground job         if(kill(-pid, sig)&lt;0){             unix_error("kill (int) error");         }     }     return; } </pre>  | <p>SIGINT를 받았다면</p> <p>fg에 있는 job에 시그널을 보낸다..</p> <p>만약 에러가 나면 처리해준다</p>   |
| <pre> void sigtstp_handler(int sig) {     pid_t pid=fgpid(jobs);      if(pid&gt;0)     {         if(kill(-pid, sig)&lt;0){             unix_error("kill (tstp) error");         }     }     return; } </pre>  | <p>SIGTSTP를 받았다면</p> <p>마찬가지로 시그널을 보낸다.</p> <p>만약 에러가 나면 처리해준다.</p>  |

### 3. 실행 결과

|  |  |
|--|--|
| <pre>[joon363@programming2 lab6]\$ make test01 ./sdriver.pl -t trace01.txt -s ./tsh -a "-p" # # trace01.txt - Properly terminate on EOF. # [joon363@programming2 lab6]\$ make rtest01 ./sdriver.pl -t trace01.txt -s ./tshref -a "-p" # # trace01.txt - Properly terminate on EOF. # [joon363@programming2 lab6]\$ make test02</pre>   | <pre>[joon363@programming2 lab6]\$ make test02 ./sdriver.pl -t trace02.txt -s ./tsh -a "-p" # # trace02.txt - Process builtin quit command. # [joon363@programming2 lab6]\$ make rtest02 ./sdriver.pl -t trace02.txt -s ./tshref -a "-p" # # trace02.txt - Process builtin quit command. # [joon363@programming2 lab6]\$ make test03</pre>   |
| trace01 tsh(위)와 tshref (아래)  | trace02 tsh(위)와 tshref (아래)  |
| <pre>[joon363@programming2 lab6]\$ make test03 ./sdriver.pl -t trace03.txt -s ./tsh -a "-p" # # trace03.txt - Run a foreground job. # tsh&gt; quit [joon363@programming2 lab6]\$ make rtest03 ./sdriver.pl -t trace03.txt -s ./tshref -a "-p" # # trace03.txt - Run a foreground job. # tsh&gt; quit [joon363@programming2 lab6]\$ █</pre>   | <pre>[joon363@programming2 lab6]\$ make test04 ./sdriver.pl -t trace04.txt -s ./tsh -a "-p" # # trace04.txt - Run a background job. # tsh&gt; ./myspin 1 &amp; [1] (5226) ./myspin 1 &amp; [joon363@programming2 lab6]\$ make rtest04 ./sdriver.pl -t trace04.txt -s ./tshref -a "-p" # # trace04.txt - Run a background job. # tsh&gt; ./myspin 1 &amp; [1] (5248) ./myspin 1 &amp; [joon363@programming2 lab6]\$ █</pre> |
| trace03 tsh(위)와 tshref(아래)   | trace04 tsh(위)와 tshref (아래)  |
| <pre>[joon363@programming2 lab6]\$ make test05 ./sdriver.pl -t trace05.txt -s ./tsh -a "-p" # # trace05.txt - Process jobs builtin command. # tsh&gt; ./myspin 2 &amp; [1] (26545) ./myspin 2 &amp; tsh&gt; ./myspin 3 &amp; [2] (26551) ./myspin 3 &amp; tsh&gt; jobs [1] (26545) Running ./myspin 2 &amp; [2] (26551) Running ./myspin 3 &amp; [joon363@programming2 lab6]\$ █</pre> | <pre>[joon363@programming2 lab6]\$ make rtest05 ./sdriver.pl -t trace05.txt -s ./tshref -a "-p" # # trace05.txt - Process jobs builtin command. # tsh&gt; ./myspin 2 &amp; [1] (5396) ./myspin 2 &amp; tsh&gt; ./myspin 3 &amp; [2] (5399) ./myspin 3 &amp; tsh&gt; jobs [1] (5396) Running ./myspin 2 &amp; [2] (5399) Running ./myspin 3 &amp; [joon363@programming2 lab6]\$ █</pre>                                     |
| trace05 tsh  | trace05 tshref   |
| <pre>[joon363@programming2 lab6]\$ make test06 ./sdriver.pl -t trace06.txt -s ./tsh -a "-p" # # trace06.txt - Forward SIGINT to foreground job. # tsh&gt; ./myspin 4 Job [1] (5523) terminated by signal 2 [joon363@programming2 lab6]\$ █</pre>   | <pre>[joon363@programming2 lab6]\$ make rtest06 ./sdriver.pl -t trace06.txt -s ./tshref -a "-p" # # trace06.txt - Forward SIGINT to foreground job. # tsh&gt; ./myspin 4 Job [1] (5585) terminated by signal 2 [joon363@programming2 lab6]\$ █</pre>   |
| trace06 tsh  | trace06 tshref   |



|   |   |
|---|---|
| <pre>[joon363@programming2 lab6]\$ make test07 ./sdriver.pl -t trace07.txt -s ./tsh -a "-p" # # trace07.txt - Forward SIGINT only to foreground job. # tsh&gt; ./myspin 4 &amp; [1] (5692) ./myspin 4 &amp; tsh&gt; ./myspin 5 Job [2] (5695) terminated by signal 2 tsh&gt; jobs [1] (5692) Running ./myspin 4 &amp; [joon363@programming2 lab6]\$ █</pre>   | <pre>[joon363@programming2 lab6]\$ make rtest07 ./sdriver.pl -t trace07.txt -s ./tshref -a "-p" # # trace07.txt - Forward SIGINT only to foreground job. # tsh&gt; ./myspin 4 &amp; [1] (5636) ./myspin 4 &amp; tsh&gt; ./myspin 5 Job [2] (5638) terminated by signal 2 tsh&gt; jobs [1] (5636) Running ./myspin 4 &amp; [joon363@programming2 lab6]\$ █</pre>   |
| trace07 tsh   | trace07 tshref  |
| <pre>[joon363@programming2 lab6]\$ make test08 ./sdriver.pl -t trace08.txt -s ./tsh -a "-p" # # trace08.txt - Forward SIGTSTP only to foreground job. # tsh&gt; ./myspin 4 &amp; [1] (26701) ./myspin 4 &amp; tsh&gt; ./myspin 5 Job [2] (26704) stopped by signal 20 tsh&gt; jobs [1] (26701) Running ./myspin 4 &amp; [2] (26704) Stopped ./myspin 5 [joon363@programming2 lab6]\$ █</pre>  | <pre>[joon363@programming2 lab6]\$ make rtest08 ./sdriver.pl -t trace08.txt -s ./tshref -a "-p" # # trace08.txt - Forward SIGTSTP only to foreground job. # tsh&gt; ./myspin 4 &amp; [1] (5797) ./myspin 4 &amp; tsh&gt; ./myspin 5 Job [2] (5801) stopped by signal 20 tsh&gt; jobs [1] (5797) Running ./myspin 4 &amp; [2] (5801) Stopped ./myspin 5 [joon363@programming2 lab6]\$ █</pre>  |
| trace08 tsh   | trace08 tshref  |
|   |   |
| <pre>[joon363@programming2 lab6]\$ make test09 ./sdriver.pl -t trace09.txt -s ./tsh -a "-p" # # trace09.txt - Process bg builtin command # tsh&gt; ./myspin 4 &amp; [1] (27008) ./myspin 4 &amp; tsh&gt; ./myspin 5 Job [2] (27013) stopped by signal 20 tsh&gt; jobs [1] (27008) Running ./myspin 4 &amp; [2] (27013) Stopped ./myspin 5 tsh&gt; bg %2 [2] (27013) ./myspin 5 tsh&gt; jobs [1] (27008) Running ./myspin 4 &amp; [2] (27013) Running ./myspin 5</pre> | <pre>[joon363@programming2 lab6]\$ make rtest09 ./sdriver.pl -t trace09.txt -s ./tshref -a "-p" # # trace09.txt - Process bg builtin command # tsh&gt; ./myspin 4 &amp; [1] (26901) ./myspin 4 &amp; tsh&gt; ./myspin 5 Job [2] (26904) stopped by signal 20 tsh&gt; jobs [1] (26901) Running ./myspin 4 &amp; [2] (26904) Stopped ./myspin 5 tsh&gt; bg %2 [2] (26904) ./myspin 5 tsh&gt; jobs [1] (26901) Running ./myspin 4 &amp; [2] (26904) Running ./myspin 5</pre> |
| trace09 tsh   | trace09 tshref  |
| <pre>[joon363@programming2 lab6]\$ make test10 ./sdriver.pl -t trace10.txt -s ./tsh -a "-p" # # trace10.txt - Process fg builtin command. # tsh&gt; ./myspin 4 &amp; [1] (27434) ./myspin 4 &amp; tsh&gt; fg %1 Job [1] (27434) stopped by signal 20 tsh&gt; jobs [1] (27434) Stopped ./myspin 4 &amp; tsh&gt; fg %1 tsh&gt; jobs [joon363@programming2 lab6]\$ █</pre>   | <pre>[joon363@programming2 lab6]\$ make rtest10 ./sdriver.pl -t trace10.txt -s ./tshref -a "-p" # # trace10.txt - Process fg builtin command. # tsh&gt; ./myspin 4 &amp; [1] (27279) ./myspin 4 &amp; tsh&gt; fg %1 Job [1] (27279) stopped by signal 20 tsh&gt; jobs [1] (27279) Stopped ./myspin 4 &amp; tsh&gt; fg %1 tsh&gt; jobs [joon363@programming2 lab6]\$ █</pre>   |
| trace10 tsh   | trace10 tshref  |

|  |  |
|--|--|
| <pre>[joon363@programming2 lab6]\$ make test11 ./sdriver.pl -t trace11.txt -s ./tsh -a "-p" # # trace11.txt - Forward SIGINT to every process # tsh&gt; ./mysplit 4 Job [1] (28486) terminated by signal 2 tsh&gt; /bin/ps a   PID TTY          STAT       TIME COMMAND  1048 pts/167    Ss          0:00 -bash  1277 pts/27     Ss+         0:00 /bin/bash --init-file=/de-server/bin/ee2b180d582a7f601fa6ecfdad8d9fd269ab1l/browser/media/shellIntegration-bash.sh  1567 tty1      Ss+         0:00 -bash  1926 pts/142    Ss+         0:00 /bin/bash --init-file=/de-server/bin/af28b32d7e553898b2a91af498b1fb666fdebe0c0minal/browser/media/shellIntegration-bash.sh</pre>   | <pre>[joon363@programming2 lab6]\$ make rtest11 ./sdriver.pl -t trace11.txt -s ./tshref -a "-p" # # trace11.txt - Forward SIGINT to every process # tsh&gt; ./mysplit 4 Job [1] (27521) terminated by signal 2 tsh&gt; /bin/ps a   PID TTY          STAT       TIME COMMAND  1048 pts/167    Ss          0:00 -bash  1277 pts/27     Ss+         0:00 /bin/bash --init-file=/de-server/bin/ee2b180d582a7f601fa6ecfdad8d9fd269ab1l/browser/media/shellIntegration-bash.sh  1567 tty1      Ss+         0:00 -bash  1926 pts/142    Ss+         0:00 /bin/bash --init-file=/de-server/bin/af28b32d7e553898b2a91af498b1fb666fdebe0c0minal/browser/media/shellIntegration-bash.sh</pre>   |
| <p>trace11 tsh</p> <p>11, 12, 13은 출력이 너무 길어 잘랐다.</p>   | <p>trace11 tshref</p>  |
| <pre>[joon363@programming2 lab6]\$ make test12 ./sdriver.pl -t trace12.txt -s ./tsh -a "-p" # # trace12.txt - Forward SIGTSTP to every process # tsh&gt; ./mysplit 4 Job [1] (29429) stopped by signal 20 tsh&gt; jobs [1] (29429) Stopped ./mysplit 4 tsh&gt; /bin/ps a   PID TTY          STAT       TIME COMMAND  1048 pts/167    Ss          0:00 -bash  1277 pts/27     Ss+         0:00 /bin/bash --init-file=/de-server/bin/ee2b180d582a7f601fa6ecfdad8d9fd269ab1l/browser/media/shellIntegration-bash.sh  1567 tty1      Ss+         0:00 -bash  1926 pts/142    Ss+         0:00 /bin/bash --init-file=/de-server/bin/af28b32d7e553898b2a91af498b1fb666fdebe0c0minal/browser/media/shellIntegration-bash.sh</pre> | <pre>[joon363@programming2 lab6]\$ make rtest12 ./sdriver.pl -t trace12.txt -s ./tshref -a "-p" # # trace12.txt - Forward SIGTSTP to every process # tsh&gt; ./mysplit 4 Job [1] (29321) stopped by signal 20 tsh&gt; jobs [1] (29321) Stopped ./mysplit 4 tsh&gt; /bin/ps a   PID TTY          STAT       TIME COMMAND  1048 pts/167    Ss          0:00 -bash  1277 pts/27     Ss+         0:00 /bin/bash --init-file=/de-server/bin/ee2b180d582a7f601fa6ecfdad8d9fd269ab1l/browser/media/shellIntegration-bash.sh  1567 tty1      Ss+         0:00 -bash  1926 pts/142    Ss+         0:00 /bin/bash --init-file=/de-server/bin/af28b32d7e553898b2a91af498b1fb666fdebe0c0minal/browser/media/shellIntegration-bash.sh</pre> |
| <p>trace12 tsh</p>   | <p>trace12 tshref</p>  |
| <pre>[joon363@programming2 lab6]\$ make test13 ./sdriver.pl -t trace13.txt -s ./tsh -a "-p" # # trace13.txt - Restart every stopped process # tsh&gt; ./mysplit 4 Job [1] (29827) stopped by signal 20 tsh&gt; jobs [1] (29827) Stopped ./mysplit 4 tsh&gt; /bin/ps a   PID TTY          STAT       TIME COMMAND  1048 pts/167    Ss          0:00 -bash  1277 pts/27     Ss+         0:00 /bin/bash --init-file=/de-server/bin/ee2b180d582a7f601fa6ecfdad8d9fd269ab1l/browser/media/shellIntegration-bash.sh  1567 tty1      Ss+         0:00 -bash  1926 pts/142    Ss+         0:00 /bin/bash --init-file=/de-server/bin/af28b32d7e553898b2a91af498b1fb666fdebe0c0minal/browser/media/shellIntegration-bash.sh</pre>    | <pre>[joon363@programming2 lab6]\$ make rtest13 ./sdriver.pl -t trace13.txt -s ./tshref -a "-p" # # trace13.txt - Restart every stopped process # tsh&gt; ./mysplit 4 Job [1] (29606) stopped by signal 20 tsh&gt; jobs [1] (29606) Stopped ./mysplit 4 tsh&gt; /bin/ps a   PID TTY          STAT       TIME COMMAND  1048 pts/167    Ss          0:00 -bash  1277 pts/27     Ss+         0:00 /bin/bash --init-file=/de-server/bin/ee2b180d582a7f601fa6ecfdad8d9fd269ab1l/browser/media/shellIntegration-bash.sh  1567 tty1      Ss+         0:00 -bash  1926 pts/142    Ss+         0:00 /bin/bash --init-file=/de-server/bin/af28b32d7e553898b2a91af498b1fb666fdebe0c0minal/browser/media/shellIntegration-bash.sh</pre>    |
| <p>trace13 tsh</p>   | <p>trace13 tshref</p>  |



|   |   |
|---|---|
| <pre>[joon363@programming2 lab6]\$ make test14 ./sdriver.pl -t trace14.txt -s ./tsh -a "-p" # # trace14.txt - Simple error handling # tsh&gt; ./bogus ./bogus: Command not found tsh&gt; ./myspin 4 &amp; [1] (12605) ./myspin 4 &amp; tsh&gt; fg fg command requires PID or %jobid argument tsh&gt; bg bg command requires PID or %jobid argument tsh&gt; fg a fg: argument must be a PID or %jobid tsh&gt; bg a bg: argument must be a PID or %jobid tsh&gt; fg 9999999 (9999999): No such process tsh&gt; bg 9999999 (9999999): No such process kill (tstp) error: No such process tsh&gt; fg %2 [joon363@programming2 lab6]\$</pre>   | <pre>[joon363@programming2 lab6]\$ make rtest14 ./sdriver.pl -t trace14.txt -s ./tshref -a "-p" # # trace14.txt - Simple error handling # tsh&gt; ./bogus ./bogus: Command not found tsh&gt; ./myspin 4 &amp; [1] (12561) ./myspin 4 &amp; tsh&gt; fg fg command requires PID or %jobid argument tsh&gt; bg bg command requires PID or %jobid argument tsh&gt; fg a fg: argument must be a PID or %jobid tsh&gt; bg a bg: argument must be a PID or %jobid tsh&gt; fg 9999999 (9999999): No such process tsh&gt; bg 9999999 (9999999): No such process kill (tstp) error: No such process tsh&gt; fg %2 [joon363@programming2 lab6]\$ make test14</pre>   |
| <p>trace14 tsh</p>  | <p>trace14 tshref</p> <p>실행할 때마다 결과가 조금씩 달라진다.<br/>아마 sleep(1)때문으로 추정된다.</p>  |
| <pre>[joon363@programming2 lab6]\$ make test15 ./sdriver.pl -t trace15.txt -s ./tsh -a "-p" # # trace15.txt - Putting it all together # tsh&gt; ./bogus ./bogus: Command not found tsh&gt; ./myspin 10 Job [1] (12910) terminated by signal 2 tsh&gt; ./myspin 3 &amp; [1] (12922) ./myspin 3 &amp; tsh&gt; ./myspin 4 &amp; [2] (12926) ./myspin 4 &amp; tsh&gt; jobs [1] (12922) Running ./myspin 3 &amp; [2] (12926) Running ./myspin 4 &amp; tsh&gt; fg %1 Job [1] (12922) stopped by signal 20 tsh&gt; jobs [1] (12922) Stopped ./myspin 3 &amp; [2] (12926) Running ./myspin 4 &amp; tsh&gt; bg %3 %3: No such job tsh&gt; bg %1 [1] (12922) ./myspin 3 &amp; tsh&gt; jobs [1] (12922) Running ./myspin 3 &amp; [2] (12926) Running ./myspin 4 &amp; tsh&gt; fg %1 tsh&gt; quit [joon363@programming2 lab6]\$</pre> | <pre>[joon363@programming2 lab6]\$ make rtest15 ./sdriver.pl -t trace15.txt -s ./tshref -a "-p" # # trace15.txt - Putting it all together # tsh&gt; ./bogus ./bogus: Command not found tsh&gt; ./myspin 10 Job [1] (12817) terminated by signal 2 tsh&gt; ./myspin 3 &amp; [1] (12824) ./myspin 3 &amp; tsh&gt; ./myspin 4 &amp; [2] (12833) ./myspin 4 &amp; tsh&gt; jobs [1] (12824) Running ./myspin 3 &amp; [2] (12833) Running ./myspin 4 &amp; tsh&gt; fg %1 Job [1] (12824) stopped by signal 20 tsh&gt; jobs [1] (12824) Stopped ./myspin 3 &amp; [2] (12833) Running ./myspin 4 &amp; tsh&gt; bg %3 %3: No such job tsh&gt; bg %1 [1] (12824) ./myspin 3 &amp; tsh&gt; jobs [1] (12824) Running ./myspin 3 &amp; [2] (12833) Running ./myspin 4 &amp; tsh&gt; fg %1 tsh&gt; quit [joon363@programming2 lab6]\$</pre> |
| <p>trace15 tsh</p>  | <p>trace15 tshref</p>   |



|  |   |
|--|---|
| <pre>[joon363@programming2 lab6]\$ make test16 ./sdriver.pl -t trace16.txt -s ./tsh -a "-p" # # trace16.txt - Tests whether the shell can handle # signals that come from other processes # tsh&gt; ./mystop 2 Job [1] (13119) stopped by signal 20 tsh&gt; jobs [1] (13119) Stopped ./mystop 2 tsh&gt; ./myint 2 Job [2] (13133) terminated by signal 2 [joon363@programming2 lab6]\$</pre> | <pre>[joon363@programming2 lab6]\$ make rtest16 ./sdriver.pl -t trace16.txt -s ./tshref -a "-p" # # trace16.txt - Tests whether the shell can handle # signals that come from other processes inside # tsh&gt; ./mystop 2 Job [1] (13067) stopped by signal 20 tsh&gt; jobs [1] (13067) Stopped ./mystop 2 tsh&gt; ./myint 2 Job [2] (13080) terminated by signal 2 [joon363@programming2 lab6]\$</pre> |
| trace16 tsh  | trace16 tshref  |

-end of document-