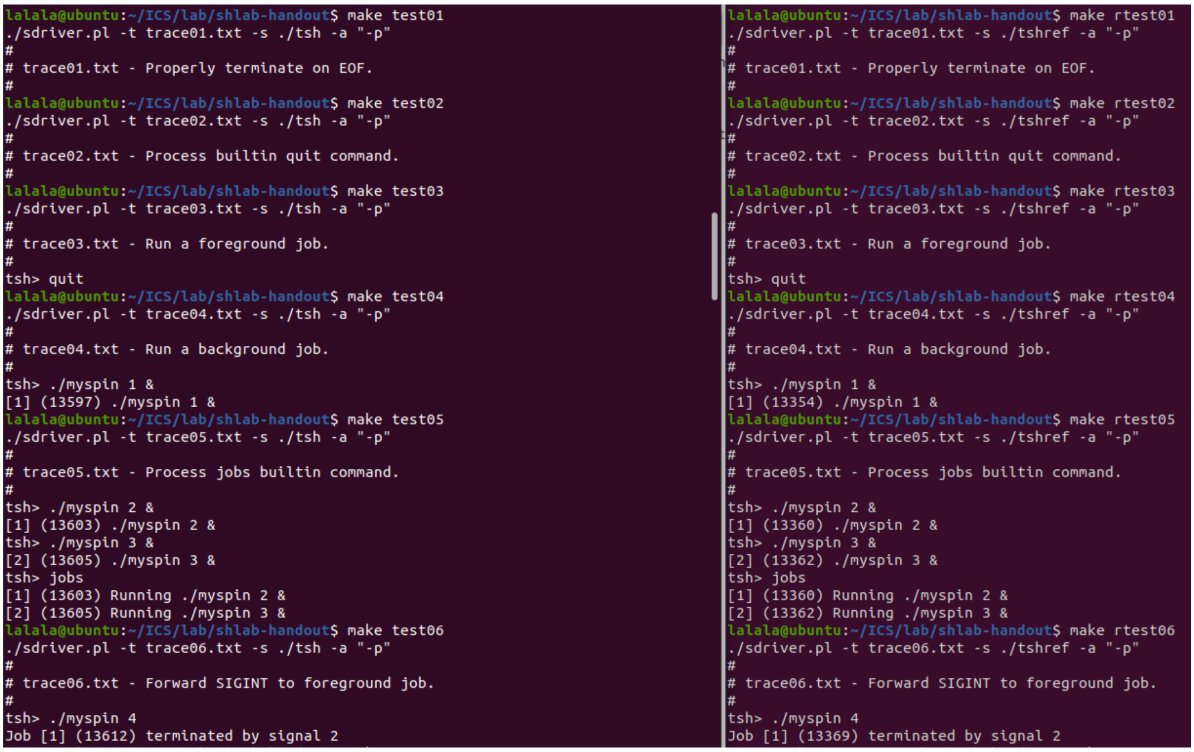
shlab-report

刘慕梵 19307130248

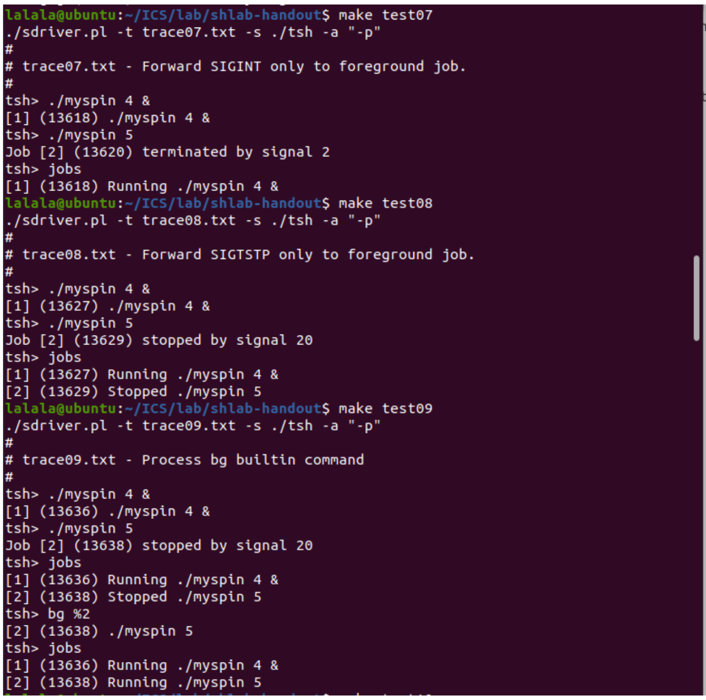
一、结果截图：

左边是tsh，右边是tshref。

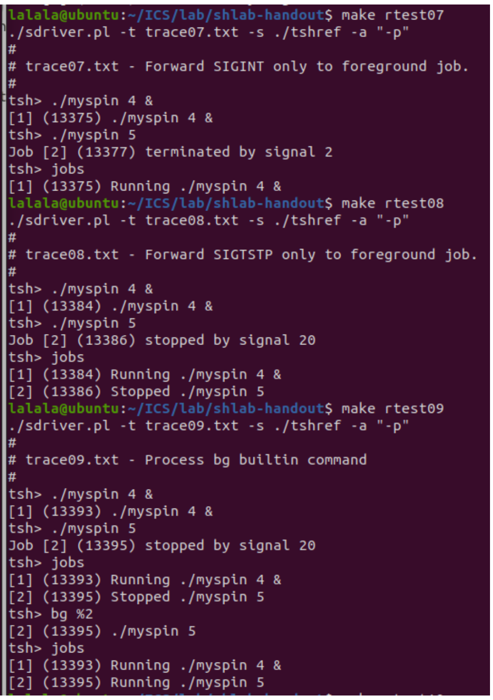


```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test01
./sdriver.pl -t trace01.txt -s ./tsh -a "-p"
#
# trace01.txt - Properly terminate on EOF.
#
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test02
./sdriver.pl -t trace02.txt -s ./tsh -a "-p"
#
# trace02.txt - Process builtin quit command.
#
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test03
./sdriver.pl -t trace03.txt -s ./tsh -a "-p"
#
# trace03.txt - Run a foreground job.
#
tsh> quit
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test04
./sdriver.pl -t trace04.txt -s ./tsh -a "-p"
#
# trace04.txt - Run a background job.
#
tsh> ./myspin 1 &
[1] (13597) ./myspin 1 &
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test05
./sdriver.pl -t trace05.txt -s ./tsh -a "-p"
#
# trace05.txt - Process jobs builtin command.
#
tsh> ./myspin 2 &
[1] (13603) ./myspin 2 &
tsh> ./myspin 3 &
[2] (13605) ./myspin 3 &
tsh> jobs
[1] (13603) Running ./myspin 2 &
[2] (13605) Running ./myspin 3 &
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test06
./sdriver.pl -t trace06.txt -s ./tsh -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (13612) terminated by signal 2
```

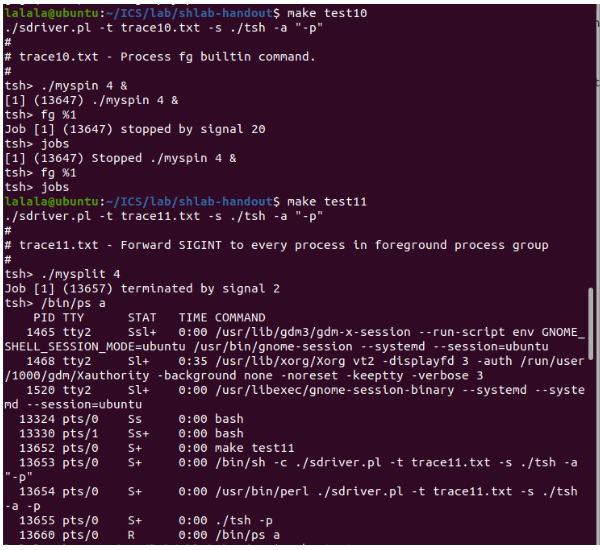
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest01
./sdriver.pl -t trace01.txt -s ./tshref -a "-p"
#
# trace01.txt - Properly terminate on EOF.
#
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest02
./sdriver.pl -t trace02.txt -s ./tshref -a "-p"
#
# trace02.txt - Process builtin quit command.
#
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest03
./sdriver.pl -t trace03.txt -s ./tshref -a "-p"
#
# trace03.txt - Run a foreground job.
#
tsh> quit
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest04
./sdriver.pl -t trace04.txt -s ./tshref -a "-p"
#
# trace04.txt - Run a background job.
#
tsh> ./myspin 1 &
[1] (13354) ./myspin 1 &
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest05
./sdriver.pl -t trace05.txt -s ./tshref -a "-p"
#
# trace05.txt - Process jobs builtin command.
#
tsh> ./myspin 2 &
[1] (13360) ./myspin 2 &
tsh> ./myspin 3 &
[2] (13362) ./myspin 3 &
tsh> jobs
[1] (13360) Running ./myspin 2 &
[2] (13362) Running ./myspin 3 &
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest06
./sdriver.pl -t trace06.txt -s ./tshref -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (13369) terminated by signal 2
```



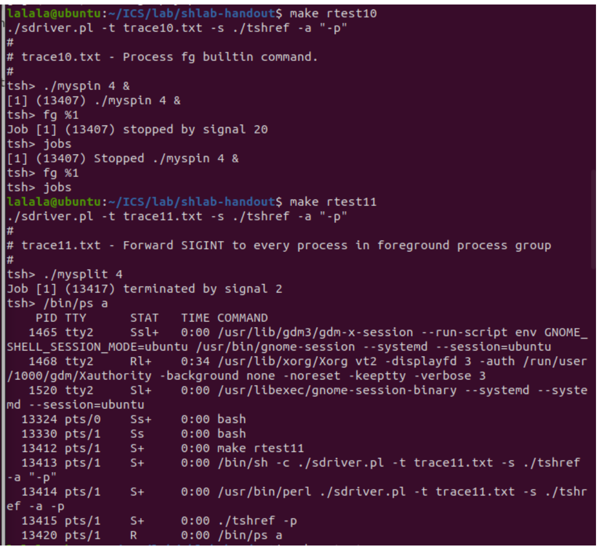
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test07
./sdriver.pl -t trace07.txt -s ./tsh -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (13618) ./myspin 4 &
tsh> ./myspin 5
Job [2] (13620) terminated by signal 2
tsh> jobs
[1] (13618) Running ./myspin 4 &
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test08
./sdriver.pl -t trace08.txt -s ./tsh -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (13627) ./myspin 4 &
tsh> ./myspin 5
Job [2] (13629) stopped by signal 20
tsh> jobs
[1] (13627) Running ./myspin 4 &
[2] (13629) Stopped ./myspin 5
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test09
./sdriver.pl -t trace09.txt -s ./tsh -a "-p"
#
# trace09.txt - Process bg builtin command
#
tsh> ./myspin 4 &
[1] (13636) ./myspin 4 &
tsh> ./myspin 5
Job [2] (13638) stopped by signal 20
tsh> jobs
[1] (13636) Running ./myspin 4 &
[2] (13638) Stopped ./myspin 5
tsh> bg %2
[2] (13638) ./myspin 5
tsh> jobs
[1] (13636) Running ./myspin 4 &
[2] (13638) Running ./myspin 5
```



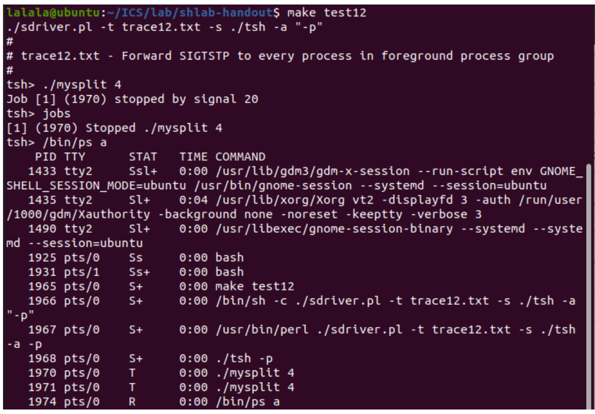
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest07
./sdriver.pl -t trace07.txt -s ./tshref -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (13375) ./myspin 4 &
tsh> ./myspin 5
Job [2] (13377) terminated by signal 2
tsh> jobs
[1] (13375) Running ./myspin 4 &
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest08
./sdriver.pl -t trace08.txt -s ./tshref -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (13384) ./myspin 4 &
tsh> ./myspin 5
Job [2] (13386) stopped by signal 20
tsh> jobs
[1] (13384) Running ./myspin 4 &
[2] (13386) Stopped ./myspin 5
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest09
./sdriver.pl -t trace09.txt -s ./tshref -a "-p"
#
# trace09.txt - Process bg builtin command
#
tsh> ./myspin 4 &
[1] (13393) ./myspin 4 &
tsh> ./myspin 5
Job [2] (13395) stopped by signal 20
tsh> jobs
[1] (13393) Running ./myspin 4 &
[2] (13395) Stopped ./myspin 5
tsh> bg %2
[2] (13395) ./myspin 5
tsh> jobs
[1] (13393) Running ./myspin 4 &
[2] (13395) Running ./myspin 5
```



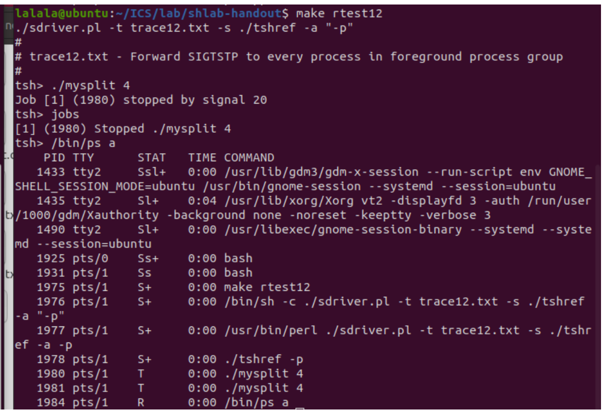
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test10
./sdriver.pl -t trace10.txt -s ./tsh -a "-p"
#
# trace10.txt - Process fg builtin command.
#
tsh> ./myspin 4 &
[1] (13647) ./myspin 4 &
tsh> fg %1
Job [1] (13647) stopped by signal 20
tsh> jobs
[1] (13647) Stopped ./myspin 4 &
tsh> fg %1
tsh> jobs
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test11
./sdriver.pl -t trace11.txt -s ./tsh -a "-p"
#
# trace11.txt - Forward SIGINT to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (13657) terminated by signal 2
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1465 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1468 tty2     Sl+    0:35 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1520 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
  13324 pts/0    Ss     0:00 bash
  13330 pts/1    Ss+    0:00 bash
  13652 pts/0    S+     0:00 make test11
  13653 pts/0    S+     0:00 /bin/sh -c ./sdriver.pl -t trace11.txt -s ./tsh -a
"-p"
  13654 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tsh
-a -p
  13655 pts/0    S+     0:00 ./tsh -p
  13660 pts/0    R      0:00 /bin/ps a
```



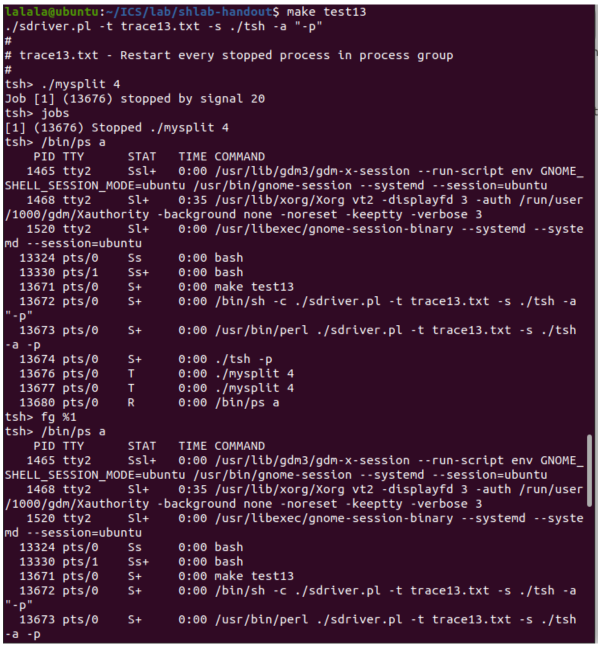
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest10
./sdriver.pl -t trace10.txt -s ./tshref -a "-p"
#
# trace10.txt - Process fg builtin command.
#
tsh> ./myspin 4 &
[1] (13407) ./myspin 4 &
tsh> fg %1
Job [1] (13407) stopped by signal 20
tsh> jobs
[1] (13407) Stopped ./myspin 4 &
tsh> fg %1
tsh> jobs
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest11
./sdriver.pl -t trace11.txt -s ./tshref -a "-p"
#
# trace11.txt - Forward SIGINT to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (13417) terminated by signal 2
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1465 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1468 tty2     Rl+    0:34 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1520 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
  13324 pts/0    Ss+    0:00 bash
  13330 pts/1    Ss     0:00 bash
  13412 pts/1    S+     0:00 make rtest11
  13413 pts/1    S+     0:00 /bin/sh -c ./sdriver.pl -t trace11.txt -s ./tshref
-a "-p"
  13414 pts/1    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tshr
ef -a -p
  13415 pts/1    S+     0:00 ./tshref -p
  13420 pts/1    R      0:00 /bin/ps a
```



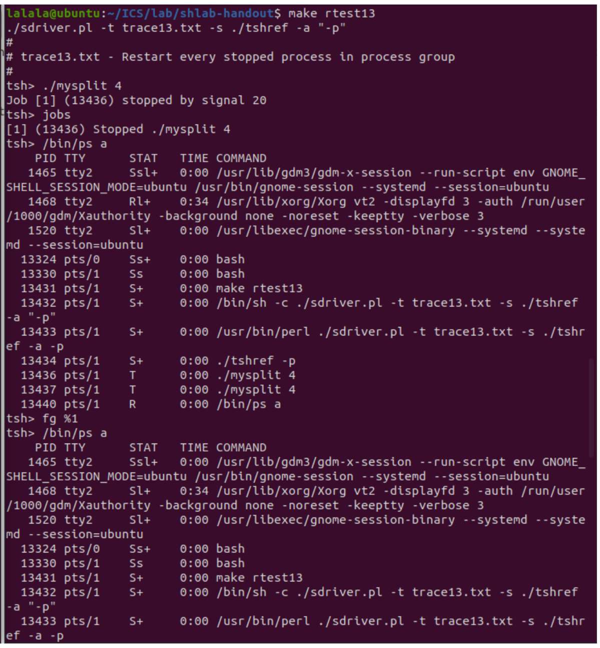
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test12
./sdriver.pl -t trace12.txt -s ./tsh -a "-p"
#
# trace12.txt - Forward SIGTSTP to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (1970) stopped by signal 20
tsh> jobs
[1] (1970) Stopped ./mysplit 4
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1433 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1435 tty2     Sl+    0:04 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1490 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
   1925 pts/0    Ss     0:00 bash
   1931 pts/1    Ss+    0:00 bash
   1965 pts/0    S+     0:00 make test12
   1966 pts/0    S+     0:00 /bin/sh -c ./sdriver.pl -t trace12.txt -s ./tsh -a
"-p"
   1967 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tsh
-a -p
   1968 pts/0    S+     0:00 ./tsh -p
   1970 pts/0    T      0:00 ./mysplit 4
   1971 pts/0    T      0:00 ./mysplit 4
   1974 pts/0    R      0:00 /bin/ps a
```



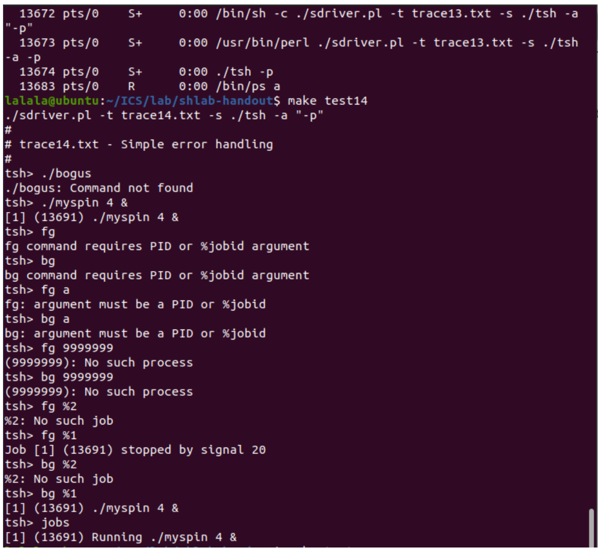
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest12
./sdriver.pl -t trace12.txt -s ./tshref -a "-p"
#
# trace12.txt - Forward SIGTSTP to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (1980) stopped by signal 20
tsh> jobs
[1] (1980) Stopped ./mysplit 4
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1433 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1435 tty2     Sl+    0:04 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1490 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
   1925 pts/0    Ss+    0:00 bash
   1931 pts/1    Ss     0:00 bash
   1975 pts/1    S+     0:00 make rtest12
   1976 pts/1    S+     0:00 /bin/sh -c ./sdriver.pl -t trace12.txt -s ./tshref
-a "-p"
   1977 pts/1    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tshr
ef -a -p
   1978 pts/1    S+     0:00 ./tshref -p
   1980 pts/1    T      0:00 ./mysplit 4
   1981 pts/1    T      0:00 ./mysplit 4
   1984 pts/1    R      0:00 /bin/ps a
```



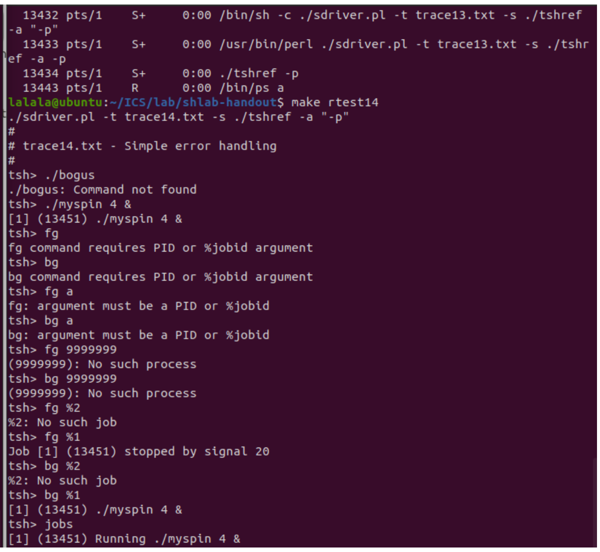
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test13
./sdriver.pl -t trace13.txt -s ./tsh -a "-p"
#
# trace13.txt - Restart every stopped process in process group
#
tsh> ./mysplit 4
Job [1] (13676) stopped by signal 20
tsh> jobs
[1] (13676) Stopped ./mysplit 4
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1465 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1468 tty2     Sl+    0:35 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1520 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
  13324 pts/0    Ss     0:00 bash
  13330 pts/1    Ss+    0:00 bash
  13671 pts/0    S+     0:00 make test13
  13672 pts/0    S+     0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tsh -a
"-p"
  13673 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh
-a -p
  13674 pts/0    S+     0:00 ./tsh -p
  13676 pts/0    T      0:00 ./mysplit 4
  13677 pts/0    T      0:00 ./mysplit 4
  13680 pts/0    R      0:00 /bin/ps a
tsh> fg %1
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1465 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1468 tty2     Sl+    0:35 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1520 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
  13324 pts/0    Ss     0:00 bash
  13330 pts/1    Ss+    0:00 bash
  13671 pts/0    S+     0:00 make test13
  13672 pts/0    S+     0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tsh -a
"-p"
  13673 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh
-a -p
```



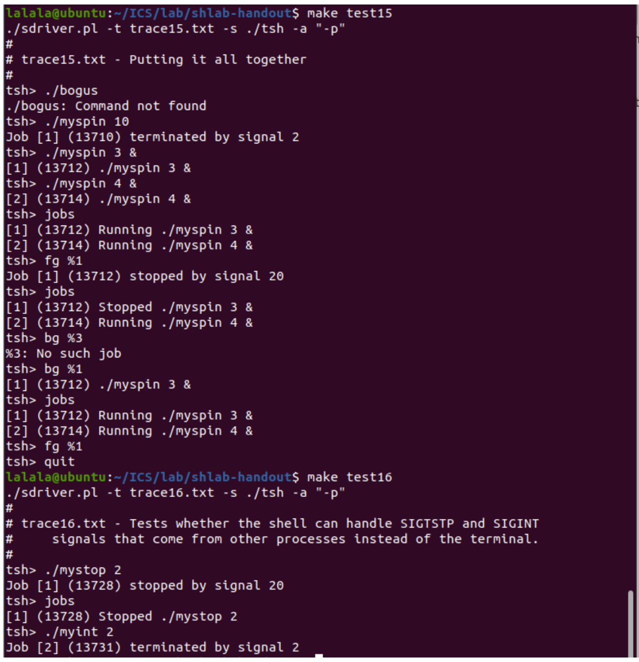
```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest13
./sdriver.pl -t trace13.txt -s ./tshref -a "-p"
#
# trace13.txt - Restart every stopped process in process group
#
tsh> ./mysplit 4
Job [1] (13436) stopped by signal 20
tsh> jobs
[1] (13436) Stopped ./mysplit 4
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1465 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1468 tty2     Rl+    0:34 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1520 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
  13324 pts/0    Ss+    0:00 bash
  13330 pts/1    Ss     0:00 bash
  13431 pts/1    S+     0:00 make rtest13
  13432 pts/1    S+     0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tshref
-a "-p"
  13433 pts/1    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshr
ef -a -p
  13434 pts/1    S+     0:00 ./tshref -p
  13436 pts/1    T      0:00 ./mysplit 4
  13437 pts/1    T      0:00 ./mysplit 4
  13440 pts/1    R      0:00 /bin/ps a
tsh> fg %1
tsh> /bin/ps a
    PID TTY      STAT   TIME COMMAND
   1465 tty2     Ssl+   0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_
SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
   1468 tty2     Sl+    0:34 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user
/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
   1520 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd --syste
md --session=ubuntu
  13324 pts/0    Ss+    0:00 bash
  13330 pts/1    Ss     0:00 bash
  13431 pts/1    S+     0:00 make rtest13
  13432 pts/1    S+     0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tshref
-a "-p"
  13433 pts/1    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshr
ef -a -p
```



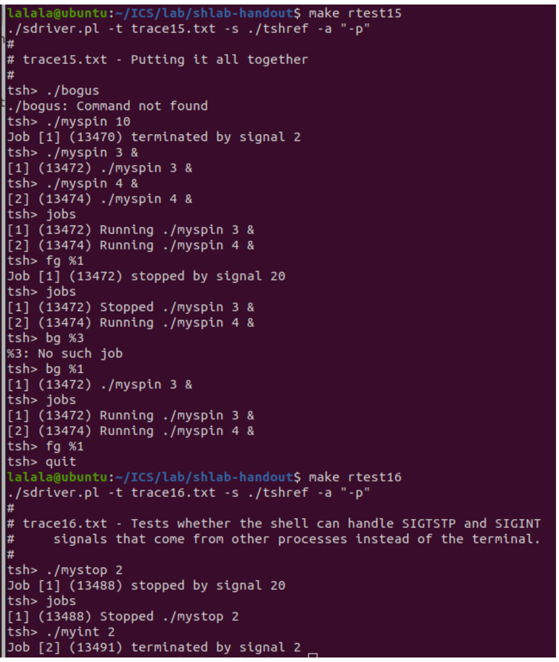
```
  13672 pts/0    S+     0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tsh -a
"-p"
  13673 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh
-a -p
  13674 pts/0    S+     0:00 ./tsh -p
  13683 pts/0    R      0:00 /bin/ps a
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test14
./sdriver.pl -t trace14.txt -s ./tsh -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (13691) ./myspin 4 &
tsh> fg
fg command requires PID or %jobid argument
tsh> bg
bg command requires PID or %jobid argument
tsh> fg a
fg: argument must be a PID or %jobid
tsh> bg a
bg: argument must be a PID or %jobid
tsh> fg 9999999
(9999999): No such process
tsh> bg 9999999
(9999999): No such process
tsh> fg %2
%2: No such job
tsh> fg %1
Job [1] (13691) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (13691) ./myspin 4 &
tsh> jobs
[1] (13691) Running ./myspin 4 &
```



```
  13432 pts/1    S+     0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tshref
-a "-p"
  13433 pts/1    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshr
ef -a -p
  13434 pts/1    S+     0:00 ./tshref -p
  13443 pts/1    R      0:00 /bin/ps a
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest14
./sdriver.pl -t trace14.txt -s ./tshref -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (13451) ./myspin 4 &
tsh> fg
fg command requires PID or %jobid argument
tsh> bg
bg command requires PID or %jobid argument
tsh> fg a
fg: argument must be a PID or %jobid
tsh> bg a
bg: argument must be a PID or %jobid
tsh> fg 9999999
(9999999): No such process
tsh> bg 9999999
(9999999): No such process
tsh> fg %2
%2: No such job
tsh> fg %1
Job [1] (13451) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (13451) ./myspin 4 &
tsh> jobs
[1] (13451) Running ./myspin 4 &
```



```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test15
./sdriver.pl -t trace15.txt -s ./tsh -a "-p"
#
# trace15.txt - Putting it all together
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 10
Job [1] (13710) terminated by signal 2
tsh> ./myspin 3 &
[1] (13712) ./myspin 3 &
tsh> ./myspin 4 &
[2] (13714) ./myspin 4 &
tsh> jobs
[1] (13712) Running ./myspin 3 &
[2] (13714) Running ./myspin 4 &
tsh> fg %1
Job [1] (13712) stopped by signal 20
tsh> jobs
[1] (13712) Stopped ./myspin 3 &
[2] (13714) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (13712) ./myspin 3 &
tsh> jobs
[1] (13712) Running ./myspin 3 &
[2] (13714) Running ./myspin 4 &
tsh> fg %1
tsh> quit
lalala@ubuntu:~/ICS/lab/shlab-handout$ make test16
./sdriver.pl -t trace16.txt -s ./tsh -a "-p"
#
# trace16.txt - Tests whether the shell can handle SIGTSTP and SIGINT
#     signals that come from other processes instead of the terminal.
#
tsh> ./mystop 2
Job [1] (13728) stopped by signal 20
tsh> jobs
[1] (13728) Stopped ./mystop 2
tsh> ./myint 2
Job [2] (13731) terminated by signal 2
```



```
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest15
./sdriver.pl -t trace15.txt -s ./tshref -a "-p"
#
# trace15.txt - Putting it all together
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 10
Job [1] (13470) terminated by signal 2
tsh> ./myspin 3 &
[1] (13472) ./myspin 3 &
tsh> ./myspin 4 &
[2] (13474) ./myspin 4 &
tsh> jobs
[1] (13472) Running ./myspin 3 &
[2] (13474) Running ./myspin 4 &
tsh> fg %1
Job [1] (13472) stopped by signal 20
tsh> jobs
[1] (13472) Stopped ./myspin 3 &
[2] (13474) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (13472) ./myspin 3 &
tsh> jobs
[1] (13472) Running ./myspin 3 &
[2] (13474) Running ./myspin 4 &
tsh> fg %1
tsh> quit
lalala@ubuntu:~/ICS/lab/shlab-handout$ make rtest16
./sdriver.pl -t trace16.txt -s ./tshref -a "-p"
#
# trace16.txt - Tests whether the shell can handle SIGTSTP and SIGINT
#     signals that come from other processes instead of the terminal.
#
tsh> ./mystop 2
Job [1] (13488) stopped by signal 20
tsh> jobs
[1] (13488) Stopped ./mystop 2
tsh> ./myint 2
Job [2] (13491) terminated by signal 2
```

二、代码及注释：

1、eval

```c
void eval(char *cmdline)
{
    char* argv[MAXLINE];
    char buf[MAXLINE];
    int bg;
    pid_t pid;

    sigset_t mask_all, mask_one, prev;
    if(sigfillset(&mask_all) < 0)//设置阻塞集合,mask_all阻塞所有信号
        unix_error("sigset error");
    if(sigemptyset(&mask_one) < 0)
        unix_error("sigset error");
    if(sigaddset(&mask_one, SIGCHLD) < 0)//mask_one阻塞SIGCHLD信号
        unix_error("sigset error");

    strcpy(buf, cmdline);
    bg = parseline(buf, argv);//得到argv，并判断是否是后台作业，得到bg
    if(argv[0] == NULL)//忽略空命令
        return;

    if(!builtin_cmd(argv))//执行内部命令，如果非内部命令执行以下代码
    {
        if(sigprocmask(SIG_BLOCK, &mask_one, &prev) < 0)//阻塞SIGCHLD信号，防止addjob时出现不一致
            unix_error("sigset error");
        if((pid = fork()) < 0)
            unix_error("fork error");
        if(pid == 0)//子进程
        {
            if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)//解除子进程对SIGCHLD的阻塞
                unix_error("sigset error");
            if(execve(argv[0], argv, environ) < 0)//调用可执行程序
            {
                printf("%s: Command not found\n",argv[0]);//出错
                exit(0);//子进程退出结束
            }
        }
        //主进程
        if(sigprocmask(SIG_BLOCK, &mask_all, NULL) < 0)//阻塞所有信号
            unix_error("sigset error");
        if(setpgid(pid, 0) < 0)//将该进程单独添加到一个进程组
            unix_error("setpid group error");
        addjob(jobs, pid, bg + 1, buf);//添加到工作集
        if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)//解除阻塞
            unix_error("sigset error");

        if(!bg)//前台作业，显式等待作业结束
            waitfg(pid);
        else
            printf("[%d] (%d) %s",pid2jid(pid), pid, buf);//后台作业，打印相关信息
    }
    return;
}
```

2、builtin cmd

```
int builtin_cmd(char **argv)
{
    sigset_t mask, prev;
    if(sigemptyset(&mask) < 0)//设置阻塞集合
        unix_error("sigset error");
    if(sigaddset(&mask, SIGCHLD) < 0)//设置阻塞集合
        unix_error("sigset error");
    if(!strcmp(argv[0], "quit"))//quit command
    {
        for(int i = 0; i < MAXJOBS; i++)
            if(jobs[i].pid)
                kill(-jobs[i].pid, SIGKILL);//结束所有子进程
        exit(0);//退出shell
    }
    if(!strcmp(argv[0], "jobs"))//jobs command
    {
        if(sigprocmask(SIG_BLOCK, &mask, &prev) < 0)//在调用listjobs时阻塞SIGCHLD信号,防止调用过程中接收到SIGCHLD信号而发生不一致
            unix_error("sigset error");
        listjobs(jobs);
        if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)
            unix_error("sigset error");
        return 1;
    }
    if(!strcmp(argv[0], "fg") || !strcmp(argv[0], "bg"))//fg bg command
    {
        do_bgfg(argv);//执行处理bg、fg的函数
        return 1;
    }
    return 0;     /* not a builtin command */
}
```

3、do_bgfg

```
void do_bgfg(char **argv)
{
    pid_t pid = 0;//fg或bg作用的pid
    struct job_t *job;//相应pid的job
    sigset_t mask, prev;
    if(argv[1] == NULL)//如果第二个参数为空，输出提示
    {
        printf("%s command requires PID or %%jobid argument\n", argv[0]);
        return;
    }
    if(argv[1][0] == '%')//如果是jid
    {
        int jid = atoi(argv[1] + 1);//得到jid
        if(jid == 0)//如果参数有错误，打印信息
        {
            printf("%s: argument must be a PID or %%jobid\n",argv[0]);
            return;
        }
        job = getjobjid(jobs, jid);//得到相应的job
```

```c
        if(job == NULL)//jid不存在
        {
            printf("%s: No such job\n", argv[1]);
            return;
        }
        pid = job->pid;
    }
    else
    {
        pid = atoi(argv[1]);//得到pid
        if(pid == 0)//如果参数有错误，打印信息
        {
            printf("%s: argument must be a PID or %%jobid\n",argv[0]);
            return;
        }
        job = getjobpid(jobs, pid);//得到相应的job
        if(job == NULL)//pid不存在
        {
            printf("(%s): No such process\n", argv[1]);
            return;
        }
    }
    if(sigemptyset(&mask) < 0)//设置阻塞集合
        unix_error("sigset error");
    if(sigaddset(&mask, SIGCHLD) < 0)//设置阻塞集合
        unix_error("sigset error");
    if(sigprocmask(SIG_BLOCK, &mask, &prev) < 0)//在发送SIGCONT时阻塞SIGCHLD信号,防止调用过程中接收到SIGCHLD信号而发生不一致
        unix_error("sigset error");
    if(kill(-pid, SIGCONT) < 0)//发送SIGCONT信号
        unix_error("kill error");
    if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)
        unix_error("sigset error");
    if(!strcmp(argv[0], "bg"))//如果是bg命令
    {
        job->state = BG;//修改状态为BG
        printf("[%d] (%d) %s",job->jid, pid, job->cmdline);//打印相应信息
    }
    else//fg命令
    {
        job->state = FG;//修改状态为fg
        waitfg(pid);//等待运行结束
    }
    return;
}
```

4、waitfg

```c
void waitfg(pid_t pid)
{
    sigset_t mask, prev;

    if(sigemptyset(&mask) < 0)//设置mask
        unix_error("sigset error");
    if(sigaddset(&mask, SIGCHLD) < 0)
        unix_error("sigset error");
    if(sigprocmask(SIG_BLOCK, &mask, &prev) < 0)//阻塞SIGCHLD信号
```

```
        unix_error("sigset error");

    while(fgpid(jobs) == pid)//循环等待前台进程结束
        sigsuspend(&prev);

    if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)//还原调用前的阻塞集合
        unix_error("sigset error");

    return;
}
```

5、sigchld_handler

```
void sigchld_handler(int sig)
{
    int olderrno = errno;//保存之前的错误信息
    sigset_t mask, prev;
    int status;
    pid_t pid;

    if(sigfillset(&mask) < 0)//设置阻塞集合
        unix_error("sigset error");

    while((pid = waitpid(-1, &status, WNOHANG | WUNTRACED)) > 0)//循环检查所有终止或停止的子进程
    {
        if(sigprocmask(SIG_BLOCK, &mask, &prev) < 0)//阻塞所有信号
            unix_error("sigset error");
        if(WIFEXITED(status))//子进程正常结束退出，删除其job
            deletejob(jobs, pid);
        else if(WIFSIGNALED(status))//因为未捕获的信号结束，打印信息并删除job
        {
            printf("Job [%d] (%d) terminated by signal %d\n",pid2jid(pid), pid, WTERMSIG(status));
            deletejob(jobs, pid);
        }
        else if(WIFSTOPPED(status))//子进程停止
        {
            printf("Job [%d] (%d) stopped by signal %d\n",pid2jid(pid), pid, WSTOPSIG(status));
            struct job_t* job = getjobpid(jobs, pid);
            job->state = ST;
        }
        if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)//恢复阻塞集合
            unix_error("sigset error");
    }
    if(errno != olderrno && errno != ECHILD)//检查waitpid返回负值的原因是不是因为没有终止或停止子进程了
        unix_error("waitpid error");

    errno = olderrno;//恢复之前的错误信息
    return;
}
```

6、sigint_handler

```c
void sigint_handler(int sig)
{
    int olderrno = errno;//保存之前的错误信息
    sigset_t mask, prev;
    pid_t pid = fgpid(jobs);//找到前台进程

    if(pid > 0)//如果存在前台作业
    {
        if(sigemptyset(&mask) < 0)//设置mask
            unix_error("sigset error");
        if(sigaddset(&mask, SIGCHLD) < 0)
            unix_error("sigset error");
        if(sigprocmask(SIG_BLOCK, &mask, &prev) < 0)//阻塞SIGCHLD信号
            unix_error("sigset error");
        if(kill(-pid, SIGINT) < 0)//发送SIGINT信号
            unix_error("kill error");
        if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)//恢复阻塞集合
            unix_error("sigset error");
    }

    errno = olderrno;//恢复之前的错误信息
    return;
}
```

7、sigtstp_handler

```c
void sigtstp_handler(int sig)
{
    int olderrno = errno;//保存之前的错误信息
    sigset_t mask, prev;
    pid_t pid = fgpid(jobs);//找到前台进程

    if(pid > 0)//如果存在前台作业
    {
        if(sigemptyset(&mask) < 0)//设置mask
            unix_error("sigset error");
        if(sigaddset(&mask, SIGCHLD) < 0)
            unix_error("sigset error");
        if(sigprocmask(SIG_BLOCK, &mask, &prev) < 0)//阻塞SIGCHLD信号
            unix_error("sigset error");
        if(kill(-pid, SIGTSTP) < 0)//发送SIGTSTP信号
            unix_error("kill error");
        if(sigprocmask(SIG_SETMASK, &prev, NULL) < 0)//恢复阻塞集合
            unix_error("sigset error");
    }

    errno = olderrno;//恢复之前的错误信息
    return;
}
```