M1522.000800: System Programming

L4. Shell lab: Session 2

Computer Systems and Platforms Laboratory
School of Computer Science and Engineering
Seoul National University

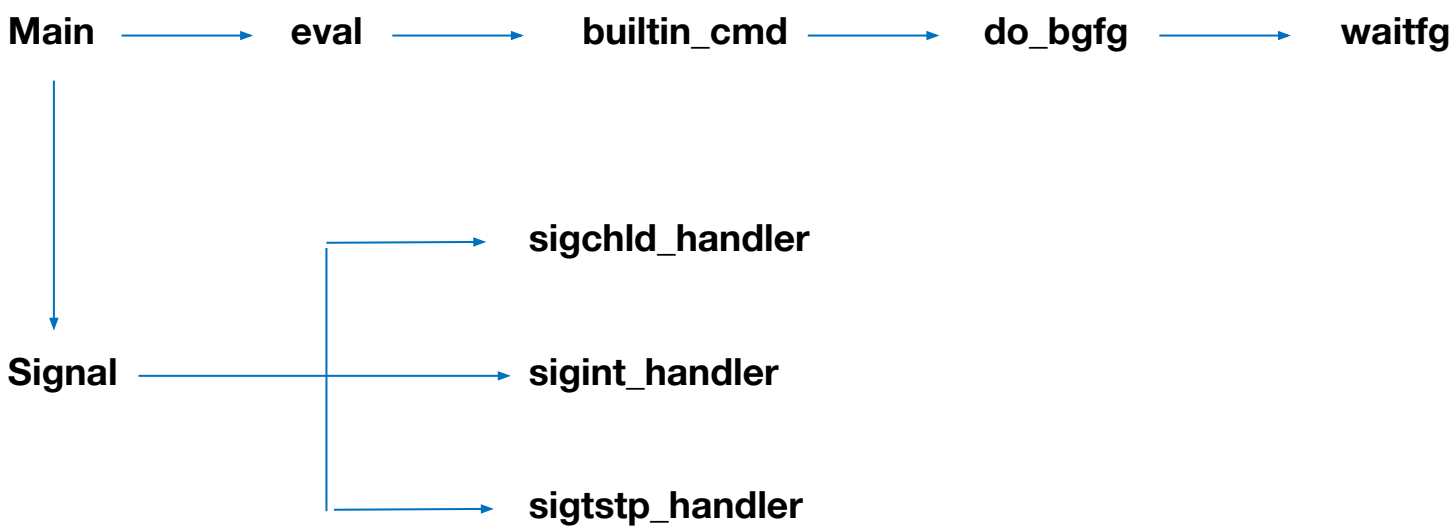
Hints

Hints

- **Read Chapter 8 (Exceptional Control Flow) in your textbook and handout!**
- Use the trace files to guide the development of your shell.
  - Starting with *trace01.txt*, make sure that your shell produces the *identical* output as the **reference shell**.
- In **eval**, the parent must use *sigprocmask* to block **SIGCHLD** signals before it forks the child and then unblock these signals
- Child process will be a member of the foreground process group by default.
  - You need to be careful to implement SIGINT handler.
- Tricky parts
  - In ***waitfg***, use a busy loop around the *sleep* function.
  - In **sigchld_handler**, use exactly one call to *waitpid*.

Hints

- 7 functions need to be implemented **eval** , **builtin_cmd** , **do_bgfg , waitfg** , **sigchld_handler , sigint_handler** , **sigtstp_handler**



Hints do_bgfg

- **do_bgfg: Handle** the *bg* and *fg* built-in commands.

Point 1	Process ID	Job ID

Point 2	FG	BG

- **Sending a signal and Change job status**

Hints sigchld_handler

- **sigchld_handler:** Catches **SIGCHLD** signals.

- **Kernel** sends a **SIGCHLD** to the shell when job terminates or stops.

  3 cases

  1) Terminated normally

  2) Terminated abnormally

  3) Stopped

QnA

QnA 1 ps

```
devel@gentoo ~/share/shell $ 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] (5451) stopped by signal 20
tsh> jobs
[1] (5451) Stopped ./mysplit 4
tsh> ps a
  PID TTY      STAT   TIME COMMAND
 2166 tty7     Rsl+   0:36 /usr/bin/X -nolisten tcp -br -deferglyphs 16 vt07 -auth /var/r
un/slim.auth
 2191 tty1     Ss+    0:00 /sbin/agetty 38400 tty1 linux
 2192 tty2     Ss+    0:00 /sbin/agetty 38400 tty2 linux
 2193 tty3     Ss+    0:00 /sbin/agetty 38400 tty3 linux
 2194 tty4     Ss+    0:00 /sbin/agetty 38400 tty4 linux
 2195 tty5     Ss+    0:00 /sbin/agetty 38400 tty5 linux
 2196 tty6     Ss+    0:00 /sbin/agetty 38400 tty6 linux
 2507 pts/0    Ss     0:00 bash
 5434 pts/0    T      0:00 ./mystop 1
 5447 pts/0    S+     0:00 make rtest12
 5448 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tshref -a -p
 5449 pts/0    S+     0:00 ./tshref -p
 5451 pts/0    T      0:00 ./mysplit 4
 5452 pts/0    T      0:00 ./mysplit 4
 5455 pts/0    R      0:00 ps a
devel@gentoo ~/share/shell $ 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] (5462) stopped by signal 19
tsh> jobs
[1] (5462) Stopped ./mysplit 4
tsh> ps a
  PID TTY      STAT   TIME COMMAND
 2166 tty7     Ssl+   0:36 /usr/bin/X -nolisten tcp -br -deferglyphs 16 vt07 -auth /var/r
un/slim.auth
 2191 tty1     Ss+    0:00 /sbin/agetty 38400 tty1 linux
 2192 tty2     Ss+    0:00 /sbin/agetty 38400 tty2 linux
 2193 tty3     Ss+    0:00 /sbin/agetty 38400 tty3 linux
 2194 tty4     Ss+    0:00 /sbin/agetty 38400 tty4 linux
 2195 tty5     Ss+    0:00 /sbin/agetty 38400 tty5 linux
 2196 tty6     Ss+    0:00 /sbin/agetty 38400 tty6 linux
 2507 pts/0    Ss     0:00 bash
 5434 pts/0    T      0:00 ./mystop 1
 5458 pts/0    S+     0:00 make test12
 5459 pts/0    S+     0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tsh -a -p
 5460 pts/0    R+     0:02 ./tsh -p
 5462 pts/0    T      0:00 ./mysplit 4
 5463 pts/0    T      0:00 ./mysplit 4
 5466 pts/0    R      0:00 ps a
```

- STAT : Process status.
  - R : Active status.
  - S : Sleep.
  - T : Stop.
- Process status can be different
- ./mysplit status is important.

  Ex: Both T, so it is okay

QnA 2 kill

General Overview of Unix Shells (Cont.)

- Examples of built-in commands supporting job control.
  - *jobs*: List the running and stopped background jobs.
  - *bg <job>* : Change a stopped background job to a running background job.
  - *fg <job>* : Change a stopped or running background job to a running in the foreground.
  - *kill <job>* : Terminate a job.

The *tsh* Specification (Cont.)

- Do not implement.

- ***tsh*** should support the following built-in commands.
  - *quit* : terminates the shell.
  - *jobs* : lists all background jobs.
  - *bg <PID or JID>* : restarts <PID or JID> by sending it a SIGCONT signal, and then runs it in the background.
  - *Fg <PID or JID>* : restarts <PID or JID> by sending it a SIGCONT signal, and then runs it in the foreground.

```
tsh> quit
tux /home/dongkwan/
```

```
tsh> jobs
[1] (12278) Running ./myspin 100 &
[2] (12280) Running ./myspin 200 &
[3] (12281) Running ./myspin 300 &
```

- ***tsh*** should reap all of its zombie children.

```
tsh> ./myspin 100 &
[1] (13034) ./myspin 100 &
tsh> jobs
[1] (13034) Running ./myspin 100 &
tsh> fg %1
```

QnA 3 pipe(|)

- Do not implement.
- You need to implement (>) file redirection
- Trace file 17, 18 will check this.

```
tsh> ls > file
tsh> cat file
Makefile
README
file
myint
myint.c
myspin
myspin.c
mysplit
mysplit.c
mystop
```

Thank you
Q & A