CSCI 3150 Tutorial

Assignment One - Part IV

HUANG Qun

SHB 120

`qhuang@cse.cuhk.edu.hk`

8/10/2012 – 12/10/2012

Outline

- Job Control
- Q & A

Job Control

Suspending a Job

- Job: Everything invoked from a single command line input
- A job may contain multiple processes
- Suspension: Invoked by Ctrl-Z (SIGTSTP is sent to the processes)

Capturing Suspending in Parent Process

- The parent should capture the stop of child processes
- Do you remember the *waitpid()* function?

```
// waitpid - wait for process to change state
pid_t waitpid(pid_t pid, int *status, int options);
```

The value of *options* is an OR of zero or more of:

- WNOHANG - return immediately if no child has exited
- WUNTRACED - also return if a child has stopped
- WCONTINUED[1] - also return if a stopped child has been resumed by delivery of SIGCONT

So, WUNTRACED is needed

[1] only after Linux 2.6.10

Job Control

Checking the Status

- If *status* is not NULL, waitpid() store status information in it.
- There are many macros to check it (See the man page for details).

Example

```
/* in the parent process */
/* suppose we want to wait process with pid */
int status;
if (waitpid(pid, &status, WUNTRACED) != pid) {/* error
     handling */}
else {
    if (WIFSTOPPED(status)) {
        /* The child is stopped */
    }
    else { /* ... */}
}
```

Job Control

Maintaining Job List

- You should store the suspended job information
- Design your data structure: array or linked list?
- REMEMBER: suspending jobs should be sorted, and the smallest job number is always one

Resuming Execution of A Job: *fg* Build-in Command

```
// kill - send signal to process
int kill(pid_t pid, int sig);
```

How to use this function?

- called by built-in command fg
- You (should) have stored the PIDs of processes in the job
- Retrieve the job information
- Send SIGCONT to all the PIDs

```
for (i=0; i<job.numProcess; i++)
    if ( kill(job.pid[i], SIGCONT) == -1 )
        {/* Error handling */}
```

After Sending the Signal

- Remove the job from your data structure of suspending jobs
- Call *waitpid()* again ...

Job Control

The *jobs* Build-in Command

- Lists all suspended jobs
- Just easy looping all current suspending jobs and print them out

Job Control

The *exit* Build-in Command

REMEMBER: when there are suspending jobs, the shell should not exit.

- when invoking *exit* build-in command, check current suspending jobs

Q & A