Homework

- Implement following operations on your shell.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| Execute a user program | Execution command should resemble bash shell. Ex> "./hello" : execute hello in current directory "/bin/ls" : execute "/bin/ls"
| process | Report a snapshot of the current process Ref> ps |
| reap “PID” | Send a kill signal to a process Ref> kill EX> kill 3556 : kill the process which has pid 3556 |
| CMD * | Run a background program Ref > CMD & EX > ./hello * > run hello in background |
Homework

- Restriction
  - No zombie process is allowed
  - Wait for child process

```
int main(void)
{
    ...
    signal(SIGCHLD, handler);
    ...
}
```

The only information in a signal is its ID and the fact that it arrived.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Default Action</th>
<th>Corresponding Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SIGINT</td>
<td>Terminate</td>
<td>Interrupt from keyboard (ctrl-c)</td>
</tr>
<tr>
<td>9</td>
<td>SIGKILL</td>
<td>Terminate</td>
<td>Kill program (cannot override or ignore)</td>
</tr>
<tr>
<td>11</td>
<td>SIGSEGV</td>
<td>Terminate &amp; Dump</td>
<td>Segmentation violation</td>
</tr>
<tr>
<td>14</td>
<td>SIGALRM</td>
<td>Terminate</td>
<td>Timer signal</td>
</tr>
<tr>
<td>17</td>
<td>SIGCHLD</td>
<td>Ignore</td>
<td>Child stopped or terminated</td>
</tr>
</tbody>
</table>

```
void handler(int sig) {
    pid_t pid;
    int stat;
    while ((pid = waitpid(-1, &stat, WNOHANG)) > 0);
    return;
}
```
Example

- ps -> kill process

```
proshb@proshb-vm:~$ ps
PID TTY      TIME CMD
2686 pts/1  00:00:01 bash
12703 pts/1  01:13:00 zombie
12704 pts/1  00:00:00 zombie <defunct>
13143 pts/1  00:00:00 ps
doosh@dell:~$ kill 12703
[1]+ Terminated ./zombie
proshb@proshb-vm:~$`
```
Behavior of Background Process

- `.hello & (Hello world)`

```
proshb@proshb-vm:~/test$ ./hello &
[1] 13199
proshb@proshb-vm:~/test$ Hello world
```

It does not matter which comes first

```
proshb@proshb-vm:~/test$ ./hello &
[1] 13199
proshb@proshb-vm:~/test$ Hello world
[1]+ Done ./hello
proshb@proshb-vm:~/test$
```
Behavior of Background Process

- ./loop & (Infinite loop)

```
proshb@proshb-vm:~$ ./loop &
[1] 13231
proshb@proshb-vm:~$
proshb@proshb-vm:~$
proshb@proshb-vm:~$
proshb@proshb-vm:~$
proshb@proshb-vm:~$
```
Note

- If your code has some changes to requirement, please note it on the contents of e-mail.

- Submit: swe3019@csl.skku.edu
- Due: 4.16(Tue) 23:59:59