` In-class practice #2： Before 21:30

To understand *signal* process by using wait() and waitpid().
Please modify the Daytime client/server programs to be the Echo client/server and to achieve the following check points:

- Check point 1: (30%)

Client can create **n=5** connections with server simultaneously
with port=9877.
1. Show the connected result on the screen at server side.
2. Use one of the connected clients to test the echo function.

- Check point 2: (30%)

To show Echo server with wait function cannot solve zombie problem completely.
1. Use Ctrl+D to send the terminal EOF character
2. Show terminated process ID (printf)
3. Show the zombie processes on the screen (ps command)

- Check point 3: (30%)

To show echo server with waitpid function can solve zombie problem completely (with handling SIGCHID signal)
1. Use Ctrl+D to send the terminal EOF character
2. Show terminated processID (printf)
3. Show the zombie processes on the screen (ps command)

- Check point 4: (10%)

To deal with the EINTR signal for *accept*

Time before 21:30,寫不完的下禮拜再補交(8折)

include file list:

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <strings.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <errno.h>
```

Hint:

```
waitpid(-1, &stat, WNOHANG)
//-1: wait for any child process to terminate
//WNOHANG: tell kernel not to block if there are no terminated children
```