CS220 - Computer System II
Assignment 13 (100 points)

Due: 12/09/2016, 11:59pm

1 Question 1 (100 points)

The below program implements a simple version of a shell program that runs programs that accept 0 arguments:

```c
#include <stdio.h>
#include <stdlib.h>
#include<string.h>
#include <errno.h>

int main()
{
  char line[1024];
  int pid, i;
  char *args[] = {&line, 0};
  while(1) {
    printf("Hello!!>");
    if(!fgets(line, 1023, stdin)) {
      break;
    }
    if(strcmp(line, "exit\n") == 0) break;

    for(i = 0; i< strlen(line); i++) {
      if(line[i] == '\n') line[i] = '\0';
    }
    pid = fork();
    if(pid == 0) { /* This is the child */
      execvp(line, args);
      fprintf(stderr, "Hello!!: %s\n", strerror(errno));
      exit(errno);
    } else {
      wait(NULL);
    }
  }
  return 0;
}
```

1. Modify the shell program such that it can execute programs that accept arguments. For example, "ls -l" and "objdump -d -M intel foo > foo.disas" are both valid commands to hello! shell. (50 points)

2. Modify the shell program such that it can execute multiple commands in the same line of input. Most shell programs allow chaining of commands. Two ampersands "&&" are used as delimiter to separate commands[1]. The command on the right of the delimitter executes if and only if the command on the left of the delimiter succeeds. For example, in the below code:

```
Hello!> ls -l && ps -A
```

"ps -A" executes if and only if 'ls -l" succeeds. You are required to support such a functionality where no more than 2 commands will be supplied to the shell separated by two ampersands. The command to the right of && must execute if and only if the one on the left does not throw an error. (50 points)

2 Submitting the result

Create a directory Assn13 and store myshell.c and a Makefile to build it.

```
$ tar -cvzf assn13_submission.tar.gz ./Assn13
```

Submit assn13.tar.gz on Blackboard.

[1] A semi colon can also be used to separate commands in the same line. The difference between using semi colon and two ampersands is that when you use a semi colon, both commands execute one after the other irrespective of whether or not the first command succeeded.