National University
of Computer & Emerging Sciences
Peshawar Campus

Problem Set:	Assignment 03	**Semester**:	Spring 2013
Points:	2		
Date Set:	February 26, 2013	**Due Date**:	March 04, 2013
Course:	CS206 Operating Systems	**Instructor**:	Nauman

1. (a) Write the following code in `getids.c`:

```c
/* getppid: print a child's and its parent's process ID numbers */
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
int main(int argc, char **argv) {
   printf("my process ID is %d\n", [Call to get PID]);
   printf("my parent's process ID is %d\n", [Call to get parents PID]);
   exit(0);
}
```

   (b) Complete the code above by replacing the contents of square brackets. The `getpid()` function from `unistd.h` returns the process's PID and `getpid()` returns that of the parent.

   (c) Compile the program using the command:

```
gcc -o getids getids.c
```

   Execute using:

```
./getids
```

2. (a) Write another program to issue the `fork` system call. Complete the following code for the provided requirements.

```c
/* fork: create a new process */
#include <stdlib.h> /* needed to define exit() */
#include <unistd.h> /* needed for fork() */
#include <sys/wait.h> /* needed for wait() */
#include <stdio.h> /* needed for printf() */
int main(int argc, char **argv) {
    int pid; /* process ID */
    pid = fork();

    if (pid == -1) {
        perror("Error");
    }
    sleep(1);
    exit(0);
}
```

   (b) Complete the two cases (parent/child) in the above code. One should execute only in child process and should print "In child" and the child's process ID. The second should run only if we are in parent and print both the parent's and the child's PID.

```c
/* fork: create a new process */
#include <stdlib.h> /* needed to define exit() */
#include <unistd.h> /* needed for fork() */
#include <stdio.h> /* needed for printf() */
int main(int argc, char **argv) {

    fork();
    fork();
    fork();

    sleep(10000);
    exit(0);
}
```

In another terminal, issue the command: `ps aux | grep forkexample`
Notice how many processes are currently running. Take a screenshot of the top command's output.