Name: Jasmine Jones

Fork(): This concept is otherwise known as a system call that imitates and generates a new process. Additionally, a new child process is created which is an identical variation of the parent process.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
   pid_t p = fork();
   if(p<0){
     perror("fork has failed");
     exit(1);
   }
   printf(process_id(pid) = %d \n",getpid());
   return 0;
}
```

Exec(): This system call will replace the current system call that is running with a new program as opposed to generating and initiating a new process.

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t p = fork();

    if (p < 0) {
        perror("fork has failed");
        exit(1);
    }
    else if (p == 0) {
        printf("Child process: pid = %d\n", getpid());
        execlp("ls", "ls", "-l", NULL);
        perror("exec failed");
        exit(1);
    }
    else {
        printf("Parent process: pid = %d\n", getpid());
        wait(NULL);
    }
```

```
    return 0;
}
```