```c
/*****************************************
 *
 *               lab2a
 *
 *  This program illustrates the use of the
 *  fork system call.
 *
 *****************************************/

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <errno.h>

int main(int argc, char **argv) {
    int pid;
    int i;
    struct timespec request;

    if((pid = fork())) {
        if(pid < 0) {
            printf("Fork error: %s\n",strerror(errno));
            exit(1);
        }
        printf("Parent: child pid: %d\n",pid);
        request.tv_sec = 0;
        request.tv_nsec = 10; //it takes about 0 nanoseconds for child
process to start
        nanosleep(&request, NULL);
        for(i=0; i<100; i++) {
            printf("Parent: %d\n",i);
        }
    } else {
        pid = getppid();
        printf("Child: parent pid: %d\n",pid);
        for(i=0; i<100; i++) {
            printf("Child: %d\n",i);
        }
```

```
    }

    exit(0);
}
```

```
/****************************************
 *
 *               lab2a
 *
 *  This program illustrates the use of the
 *  execve system call.
 *
 ****************************************/

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/wait.h>

extern char **environ;

int main(int argc, char **argv) {
    int pid;
    int ret;
    int status;

    if((pid = fork())) {
        if(pid < 0) {
            printf("Fork error: %s\n",strerror(errno));
            exit(1);
        }
        printf("Wait: %d\n", wait(&status));
    } else {
        ret = execve("lab2a", argv, environ);
        if(ret < 0) {
            printf("Execve failed: %s\n", strerror(errno));
            exit(1);
```

```
        }
    }

    exit(0);
}
```

```
(base) andre@DESKTOP-UM1B7BM:/mnt/c/Users/andre/OneDrive/Systems/Lab02$ ./lab2a
Child: parent pid: 89
Parent: child pid: 90
Child: 0
Parent: 0
Child: 1
Parent: 1
Child: 2
Parent: 2
Child: 3
Parent: 3
Child: 4
Parent: 4
Child: 5
Parent: 5
Child: 6
Parent: 6
Child: 7
Parent: 7
Child: 8
Parent: 8
Child: 9
Parent: 9
Child: 10
Parent: 10
Child: 11
Parent: 11
Child: 12
Parent: 12
```