```
/*
                        PROCESS FORKING

  This program illustrates the fork() system call.

*/

#include <stdio.h>
#include <sys/ipc.h>

main()
{
        int i ;
        int x = 10 ;
        int pid1, pid2, status ;

        printf("Before forking, the value of x is %d\n", x);

        /*
           After forking, we make the parent and its two children
            increment x in different ways to illustrate that they
           have different copies of x
        */
        if ((pid1 = fork()) == 0) {

                /* First child process */
                for (i=0 ; i < 5; i++) {
                   printf("\t\t\t At first child: x= %d\n", x);
                   x= x+10;
                   sleep(1) ; /* Sleep for 1 second */
                }
        }
        else {

                /* Parent process */

                /* Create another child process */
                if ((pid2 = fork()) == 0) {

                    /* Second child process */
                     for (i=0 ; i < 5; i++) {
                        printf("\t\t\t\t\t\t At second child: x= %d\n", x);
                        x= x+20;
                        sleep(1) ; /* Sleep for 1 second */
                     }
                }
                else {

                        /* Parent process */
                        for (i=0 ; i < 5; i++) {
                                printf("At parent: x= %d\n", x);
                                x= x+5;
                                sleep(1) ; /* Sleep for 1 second */
                        }

                        /*
                            The waitpid() system call causes the parent
```

```
                        to wait for a child process with a specific pid to
                        complete its execution. The input parameter can
                        specify the PID of the child process for
                        which it has to wait.
                    */

                    waitpid(pid1, &status, 0);
                    waitpid(pid2, &status, 0);
                }
        }
}
```