Network Programming

Exec example

The following program forks off a child process. The child process calls the *execlp* function to evoke an *xterm* in which the program *test* is executed. "127.0.0.1" is passed to *test* as a command line argument. The parent process blocks itself using *wait* till the child terminates, then prints out a message and terminates.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

main(int argc, char **argv) {

  pid_t childpid, pid;
  int   stat;

  printf("forking child\n");

  if ( (childpid = fork()) == 0)  {

      if ( (execlp("xterm", "xterm", "-e", "./test", "127.0.0.1", (char *) 0)) < 0)  {
            /*print an appropriate error message here*/
            exit(1);
         }
     }

  pid = wait(&stat);
  printf("child terminated\n");
}
```

When compiling, the executable file of this program should be given the name *test* and reside in the same directory as the program above. *test* sits in an infinite loop, printing out the command line argument given to it, then sleeps for one second.
To terminate the program, type ^C (CTRL C) in the *xterm* window. Notice that this kills both *test* and the *xterm* in which it is running, because it kills the entire child process.

```
#include <stdio.h>

main(int argc, char** argv) {

  for ( ; ; ) {
      printf("%s\n", argv[1]);
      sleep (1);
  }
}
```