

wait()- program structure

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

main( int argc, char* argv[] )
{
    pid_t childPID;

    if ( ( childPID = fork() ) == 0 ) {
        /*child*/
    }
    else {
        /*parent*/
        wait ( 0 );
    }
    exit( 0 );
}
```

pipe() system call

pipe() - to create a read-write pipe that may later be used to communicate with a process we'll fork off.

SYNOPSIS

```
int pipe( pfd )  
int pfd[2];
```

PARAMETER:

pfd is an array of 2 integers, which that will be used to save the two file descriptors used to access the pipe.

RETURN VALUE:

0 - success;
-1 - error.

pipe() - structure

```
/* first, define an array to store the two file descriptors
*/
int pipes[2];

/* now, create the pipe */
int rc = pipe(pipes);
if (rc == -1) { /* pipe() failed */
    perror("pipe");
    exit(1);
}
```

If the call to *pipe()* succeeded, a pipe will be created, *pipes[0]* will contain the number of its read file descriptor, and *pipes[1]* will contain the number of its write file descriptor.