WAITPID

- The interpretation of pid in waitpid depends on its value

1. Pid == -1 – waits for any child
2. Pid > 0 – waits for child whose process ID equals pid
3. Pid == 0 – waits for child whose process group ID equals that of calling process
4. Pid < -1 – waits for child whose process group ID equals to absolute value of pid

- Waitpid helps us wait for a particular process
- It is nonblocking version of wait
- It supports job control

WNOHANG	Waitpid will not block if the child specified is not available
WUNTRACED	supports job control

```
#include    <sys/types.h>
#include    <sys/wait.h>
#include    "ourhdr.h"
Int main(void)
{
    pid_t    pid;
    int              status;
    if ( (pid = fork()) < 0)
            err_sys("fork error");
    else if (pid == 0)              /* child */
            exit(7);
if (wait(&status) != pid)
                        /* wait for child */
            err_sys("wait error");
```

```
        pr_exit(status);
                        /* and print its status */
    if ( (pid = fork()) < 0)
            err_sys("fork error");
    else if (pid == 0)              /* child */
            abort();
              /* generates SIGABRT */
if (wait(&status) != pid)
                            /* wait for child */
            err_sys("wait error");
    pr_exit(status);
                    /* and print its status */
    if ( (pid = fork()) < 0)
            err_sys("fork error");
    else if (pid == 0)              /* child */
            status /= 0;
        /* divide by 0 generates SIGFPE */
    if (wait(&status) != pid)
                            /* wait for child */
            err_sys("wait error");
        pr_exit(status);
                    /* and print its status */

    exit(0);
}
```

Source : http://elearningatria.files.wordpress.com/2013/10/cse-iv-unix-and-shell-programming-10cs44-notes.pdf