

WAIT AND WAITPID FUNCTIONS

5.3 Wait and waitpid functions

- When a child id terminated the parent is notified by the kernel by sending a SIGCHLD signal
- The termination of a child is an asynchronous event
- The parent can ignore or can provide a function that is called when the signal occurs
- The process that calls wait or waitpid can
 1. Block
 2. Return immediately with termination status of the child
 3. Return immediately with an error

```
#include <sys/wait.h>
```

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

```
pid_t wait (int *statloc);
```

```
pid_t waitpid (pid_t pid,int *statloc , int options );
```

- Statloc is a pointer to integer
- If statloc is not a null pointer ,the termination status of the terminated process is stored in the location pointed to by the argument
- The integer status returned by the two functions give information about exit status, signal number and about generation of core file
- Macros which provide information about how a process terminated

Program to demonstrate the use of the exit status

```
#include "apue.h"
```

```
#include <sys/wait.h>
```

```

Void pr_exit(int status)
{
    if (WIFEXITED(status))
        printf("normal termination, exit status = %d\n",WEXITSTATUS(status));
    else if (WIFSIGNALED(status))
        printf("abnormal termination, signal number = %d%s\n",WTERMSIG(status),
#ifdef WCOREDUMP
            WCOREDUMP(status) ? " (core file generated)" : "");
#else
            "");
#endif

    else if (WIFSTOPPED(status))
        printf("child stopped, signal number = %d\n",WSTOPSIG(status));
}

```

WIFEXITED	TRUE – if child terminated normally WEXITSTATUS – is used to fetch the lower 8 bits of argument child passed to exit or _exit
WIFSIGNALED	TRUE – if child terminated abnormally WTERMSIG – is used to fetch the signal number that caused termination WCOREDUMP – is true is core file was generated
WIFSTOPPED	TRUE – for a child that is currently stopped WSTOPSIG -- is used to fetch the signal number that caused child to stop

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