

## QWAIT(f)

## QWAIT(f)

### NAME

`qwait` – check for child process termination

### SYNOPSIS

(`qwait` = 80.; not in assembler)

**sys `qwait`**

(process ID in `r0`)

(status in `r1`)

**`pid = qwait(status)`**

**`int *status;`**

**`int pid;`**

### DESCRIPTION

*Qwait* checks for the termination of one of its child processes. If any child has died since the last *wait* or *qwait*, a valid return is generated; if there are no children, return is immediate with the error bit set (resp. with a value of `-1` returned). The normal return yields the process ID of the terminated child (in `r0`). A value of zero is returned for the process ID if no child process has terminated. In the case of several children several *qwait* calls are needed to learn of all the deaths.

If no error is indicated on return, the `r1` high byte (resp. the high byte stored into *status*) contains the low byte of the child process `r0` (resp. the argument of *exit*) when it terminated. The `r1` (resp. *status*) low byte contains the termination status of the process. See signal (II) for a list of termination statuses (signals); 0 status indicates normal termination. If the 0200 bit of the termination status is set, a core image of the process was produced by the system.

If the parent process terminates without waiting on its children, the termination message is lost.

This system call differs from *wait* in that the calling process does not delay waiting for process termination.

### SEE ALSO

*exit* (II), *fork* (II), *signal* (II), *wait* (II).

### DIAGNOSTICS

The error bit (c-bit) is set if there are no children not previously waited for. From C, a returned value of `-1` indicates an error.