wait3()

❓ A Feature Test Macro is a preprocessor definition that controls which symbols from the C library headers are exposed when including system headers like “unistd.h”, “stdio.h”...

❓ This function is not part of POSIX and is considered obsolete.

It may not be available on all systems.

Check “wait3()” availability at compile time using Feature Test Macros: “_BSD_SOURCE” or “_DEFAULT_SOURCE”.

The `wait3()` function:

- Allows non-blocking or selective waits via “options” parameter
  - E.g. use “WNOHANG” to poll childs without blocking
  - Catch stopped or continued processes with “WUNTRACED” or “WCONTINUED”
- Collects resource usage statistics of the terminated child via “rusage” parameter
- Unlike “waitpid()”, it doesn’t require PID: waits for any child
- “wait4()” is recommended over “wait3()”

It waits for any child to terminate, similar to “wait()”, but optionally gives CPU and memory usage stats. It’s included in the “sys/wait.h” system header.

```
#include <sys/types.h>  // includes "pid_t"
#include <sys/resource.h>  // includes "struct rusage"
#include <sys/wait.h>

pid_t wait3(int *wstatus, int options, struct rusage *rusage);
```

Example Usage

```
/* includes */
int main(void)
{
   pid_t pid = fork();
   int status;
   struct rusage usage;

   if (pid == 0)
   {  // child process (burn CPU)
      for (volatile int i = 0; i < 100000000; ++i);
      exit(0);
   }
   else
   {
      wait3(&status, 0, &usage);
      if (WIFEXITED(status))
      {
         printf("Child exited normally\n");
         printf("User time: %ld.%06ld sec\n", usage.ru_utime.tv_sec, usage.ru_utim
         printf("System time: %ld.%06ld sec\n", usage.ru_stime.tv_sec, usage.ru_s
      }
   }
   return (0);
}
```

Conditional use of wait3()

```
/* includes... */
#if defined(_BSD_SOURCE) || defined(_DEFAULT_SOURCE)
int use_wait3_example(void) {
    int status;
    struct rusage usage;
    pid_t pid = fork();
    if (pid == 0) {
        exit(0);
    } else {
        wait3(&status, 0, &usage);
        if (WIFEXITED(status)) {
            printf("Child exited with status: %d\n", WEXITSTATUS(status));
        }
        return (0);
    }
}
#else
int use_wait3_example(void) {
    fprintf(stderr, "wait3() not supported on this system.\n");
    return (1);
}
#endif
int main(void) {
    return (use_wait3_example());
}
```