

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <stdlib.h>
#include <time.h>
#include <sys/resource.h>
#include <sys/stat.h>

#include "wwv.h"

int main(int argc, char* argv[])
{
    int ret_val;           // Return
    int fd;                // File
    time_t now;            // Datetime Variables
    struct tm *local;
    int timeday[4];
    pid_t child1;          // Forks
    pid_t child2;

    // Gets datetime info
    time(&now);
    local = localtime(&now);

    // Gets appropriate data
    timeday[0] = local->tm_year + 1900;
    timeday[1] = local->tm_yday;
    timeday[2] = local->tm_hour;
    timeday[3] = local->tm_min;

    // tests O_NONBLOCK
    fd = open("/dev/wwv", O_WRONLY|O_NONBLOCK);
    if (fd < 0) {
        printf("Can't open device file");
        return -1;
    }

    child1 = fork();

    if (child1 > 0) {
        ret_val = ioctl(fd, WWV_TRANSMIT, &timeday);
        if (ret_val < 0) {
            printf("Parent ioctl failed\n");
            return -1;
        }
    }
    else {
        ret_val = ioctl(fd, WWV_TRANSMIT, &timeday);
        if (ret_val < 0) {
            printf("Child ioctl failed (Probably due to NONBLOCK)\n");
        }
    }

    close(fd);

    // Tests with multiple processes
    fd = open("/dev/wwv", O_WRONLY);
    if (fd < 0) {
        printf("Can't open device file");
        return -1;
    }

    if (child1 > 0) {
        printf("Parent\n");
        ret_val = ioctl(fd, WWV_TRANSMIT, &timeday);
    }
```

```
    if(ret_val < 0) {
        printf("Parent ioctl failed\n");
        return -1;
    }
}
else {
    printf("Child\n");
    ret_val = ioctl(fd, WWV_TRANSMIT, &timeday);
    if(ret_val < 0) {
        printf("Child ioctl failed\n");
        return -1;
    }
}

close(fd);

// Tests with more processes and delay
fd = open("/dev/wwv", O_WRONLY);
if (fd < 0) {
    printf("Can't open device file");
    return -1;
}

child2 = fork();

if (child1 > 0 && child2 > 0) {
    sleep(5);
    ret_val = ioctl(fd, WWV_TRANSMIT, &timeday);
    if(ret_val < 0) {
        printf("Parent ioctl failed\n");
        return -1;
    }
}
else if (child1 == 0 || child2 == 0) {
    ret_val = ioctl(fd, WWV_TRANSMIT, &timeday);
    if(ret_val < 0) {
        printf("Child ioctl failed\n");
        return -1;
    }
}
else {
    printf("ioctl failed\n");
    return -1;
}

close(fd);

return 0;
}
```