[Nanosleep()]

nanosleep() is used to **pause** (make the program sleep) for a **specific amount of time** — down to **nanoseconds** accuracy.

2 implement nanosleep() system call

1. Introduction

A **system call** is a way for programs to interact with the operating system. The `nanosleep()` system call is used in C programming to suspend the execution of a program for a specified amount of time. Unlike other delay methods, `nanosleep()` provides **nanosecond-level precision**, which makes it useful for applications that require precise timing, such as real-time systems, simulations, or hardware interactions.

⌛ 2. Purpose of `nanosleep()`

The purpose of using `nanosleep()` is to delay a process for a desired time interval with higher precision than `sleep()` or `usleep()`.

Header file: `<time.h>`

Function prototype:

```
int nanosleep(const struct timespec *req, struct timespec *rem);
```

Parameters:

`req`: A pointer to a `timespec` structure that specifies how long to sleep.

`rem`: A pointer to a `timespec` structure that, if the sleep is interrupted, will hold the remaining time.

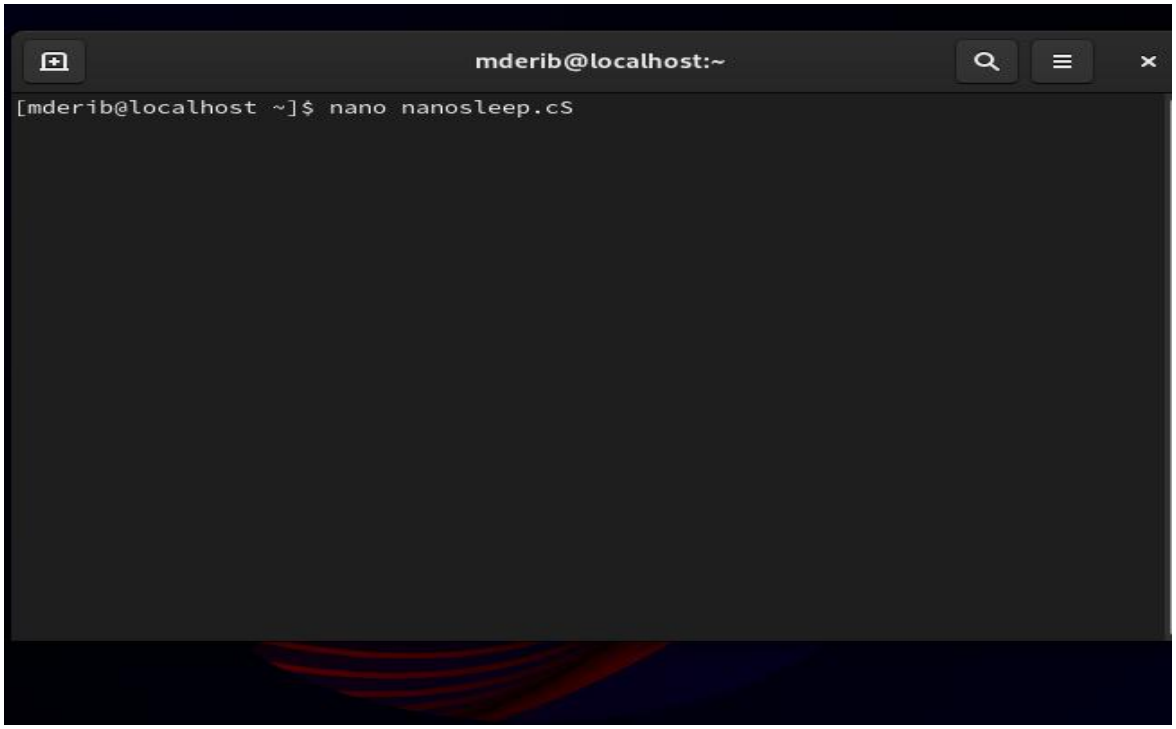
Return:

`0` on success.

`-1` on failure (e.g., if interrupted by a signal)

---

3. Source Code

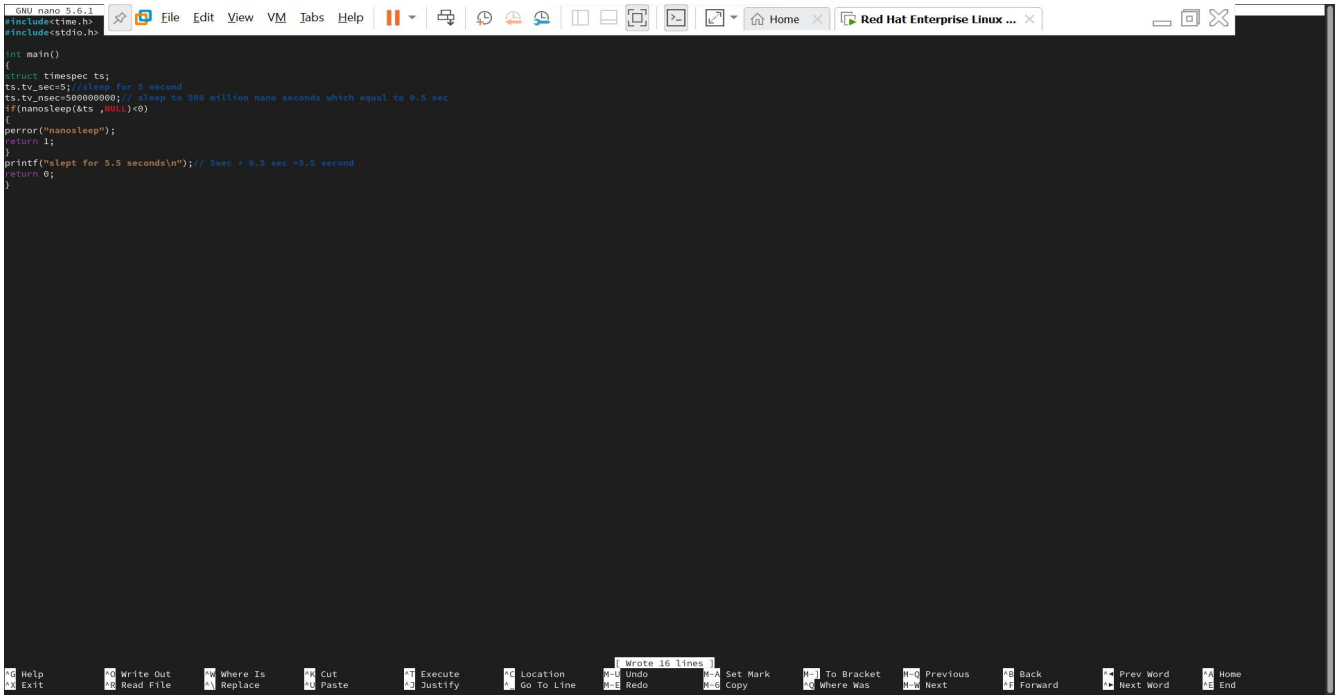



```
#include <time.h>
#include <stdio.h>
int main() {
    struct timespec ts;

    ts.tv_sec = 1;          // Sleep for 1 second
    ts.tv_nsec = 500000000; // Sleep for 0.5 second (500 million
nanoseconds)

    if (nanosleep(&ts, NULL) < 0) {
        perror("nanosleep");
        return 1;
    }

    printf("Slept for 1.5 seconds\n");
    return 0;
}
```



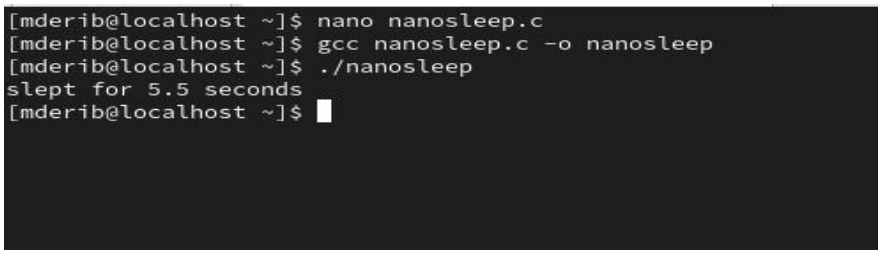
4. Compilation and Execution

Use the `gcc` compiler to compile the program:

```
gcc nanosleep.c -o nanosleep
```

Then, execute it:

```
./nanosleep
```



Expected Output:

```
Slept for 5.5 seconds
```

---

5. Conclusion

The `nanosleep()` system call allows precise control over time delays in C programs. It is an essential tool when developing applications that rely on accurate timing. This experiment demonstrates how system calls are used in real user-space programs to interact with the kernel and manage resources like time