

Implementation of System Call

System call

Steps to explain the system call(code)

Step 1:Include Necessary Headers:

We include headers for file operations, memory mapping, and standard I/O:

```
#include <iostream>
#include <fcntl.h>
#include <unistd.h>
#include <sys/mman.h>
#include <cstring>
#include <sys/stat.h>
```

Step 2:Define Constants

We set the file path and size:

```
const char* FILE_PATH = "example.txt";
const size_t FILE_SIZE = 4096;
```

Step 3: Open or Create the File

We open the file with read-write permissions, creating it if it doesn't exist

```
int fd = open(FILE_PATH, O_RDWR | O_CREAT, 0644);
if (fd == -1) {
    perror("Error opening file");
    return 1;
}
```

Step 4: Resize the File

We ensure the file is the desired size

```
if (ftruncate(fd, FILE_SIZE) == -1) {
    perror("Error resizing file");
    close(fd);
    return 1;
}
```

Step 5: Memory Map the File

We map the file into memory

```
void* addr = mmap(nullptr, FILE_SIZE, PROT_READ | PROT_WRITE, MAP_SHARED, fd,
0);
if (addr == MAP_FAILED) {
    perror("Error mapping file");
    close(fd);
    return 1; }
```

Step 6: Write to the Mapped Memory

We write a message to the memory-mapped area

```
const char* message = "Hello from memory-mapped C++!";
std::memcpy(addr, message, std::strlen(message));
```

Step 7: Flush Changes to Disk

We use `msync()` to ensure changes are written to disk

```
if (msync(addr, FILE_SIZE, MS_SYNC) == -1) {
    perror("Error syncing to disk");
} else {
    std::cout << "Successfully synced changes with msync().\n";
}
```

Step 8: Cleanup

We unmap the memory and close the file descriptor

```
if (munmap(addr, FILE_SIZE) == -1) {
    perror("Error unmapping memory");
}
close(fd);
```

Compilation and Execution

To compile and run the program:

1. Save the code to a file, e.g., `msync_example.cpp`.
2. Open a terminal and navigate to the directory containing the file.
3. Compile the program:

```
g++ -o msync_example msync_example.cpp
```

4. Run the executable:

```
./msync_example
```

5. Check the contents of `example.txt`:

```
cat example.txt
```