Lab-08: Input/Output Operations

Muhammad Shafeen

1 Introduction

This document provides a detailed overview of Input/Output (I/O) operations in Linux, including examples of using file descriptors, opening, closing, reading, and writing to files in C.

2 File Descriptors

In Linux, there are three types of file descriptors available for every process:

1. Standard Input (0)
2. Standard Output (1)
3. Standard Error (2)

Other files are assigned descriptors starting from 3 onward.

3 Opening a File

To open a file, use the `open()` system call:

Listing 1: Opening a File

```c
#include <fcntl.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

int main(int argc, char* argv[]) {
    char *path = argv[1];
    int fd = open(path, O_WRONLY | O_CREAT | O_EXCL);
    if (fd == -1) {
        printf("Error: File not created\n");
        return 1;
    }
    close(fd);
    return 0;
}
```

Compile the code using:

```
gcc demo.c -o demo
./demo createThisFile
```

4 Writing to a File

The `write()` function is used to write data to a file:

Listing 2: Writing to a File

```
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

int main(int argc, char* argv[]) {
    char *filename = argv[1];
    int fd = open(filename, O_WRONLY | O_CREAT | O_APPEND, 0666);
    if (fd == -1) {
        printf("Error: Cannot open file\n");
        return 1;
    }
    const char *data = "Hello, World!\n";
    write(fd, data, strlen(data));
    close(fd);
    return 0;
}
```

5 Reading from a File

Use the `read()` function to read data:

Listing 3: Reading from a File

```
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>

int main(int argc, char* argv[]) {
    char buffer[256];
    int fd = open(argv[1], O_RDONLY);
    if (fd == -1) {
        printf("Error: File not found\n");
        return 1;
    }
    int bytesRead = read(fd, buffer, sizeof(buffer) - 1);
    buffer[bytesRead] = '\0';
    printf("File contents:\n%s\n", buffer);
    close(fd);
    return 0;
}
```