



# Unit VIII

# (File Handling)

Lecture Slide

AS2023





Royal University of Bhutan



# File Descriptors



Royal University of Bhutan



# File Descriptors

- **What are File Descriptors?**
  - File descriptors are unique identifiers for open files in C programming. They are integers that provide a way for a program to manage multiple open files simultaneously.
  - Used by the Operating system to keep track of open files and to manage file operations
- **Significance of File Descriptors**
  - File descriptors are essential for performing input/output (I/O) operations in C. They allow programs to read data from files, write data to files, and control file operations such as opening, closing, and seeking.



Royal University of Bhutan



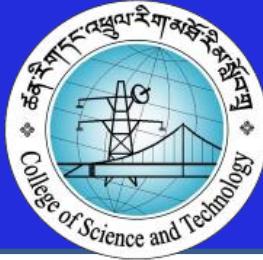
# File Descriptors

## Standard File Descriptors

- 0 (stdin) : Standard input file descriptor
- 1 (stdout) : Standard output file descriptor
- 2 (stderr) : Standard error file descriptor

## Manipulating File descriptors

- Manipulated using system calls – open, read, write, close, create functions



# Creating File Descriptors

## Opening Files

- The **open()** function is used to open a file and create a corresponding file descriptor. The open() function takes three arguments:
  - **filename:** The name of the file to open
  - **mode:** The mode in which to open the file (e.g., r for reading, w for writing, a for appending)
  - **permissions:** The permissions to set for the file (optional)

Example: `int fd = open("my_file.txt", "r");`



# Using File Descriptors for I/O Operations



## Reading Data from Files

- The **read()** function is used to read data from an opened file.
- The `read()` function takes three arguments:
  - ***fd***: The file descriptor of the file to read from
  - ***buffer***: A pointer to a buffer to store the read data
  - ***count***: The number of bytes to read

Example:

```
char buffer[1024];  
int bytes_read = read(fd,  
buffer,sizeof(buffer));
```



# Using File Descriptors for I/O Operations



## Writing Data to Files

- The **write()** function is used to write data to an open file.
- The **write()** function takes three arguments:
  - **fd**: The file descriptor of the file to write to
  - **buffer**: A pointer to the data to write
  - **count**: The number of bytes to write

Example:

```
char buffer[1024];
int bytes_written = write(fd, buffer,
sizeof(buffer));
```



# Using File Descriptors for I/O Operations



## Controlling File Operations:Seeking within Files

- The **fseek()** function is used to reposition the file pointer within an opened file. The fseek() function takes three arguments:
  - fd:** The file descriptor of the file to seek in
  - offset:** The offset from the beginning of the file to seek to
  - whence:** The origin of the offset (e.g., SEEK\_SET for the beginning of the file, SEEK\_CUR for the current position, SEEK\_END for the end of the file)

Example:

```
fseek(fd, 100, SEEK_SET);
```



# Using File Descriptors for I/O Operations



## **Closing Files:**

- The **close()** function is used to close an opened file and release its associated file descriptor.

Example:

```
close(fd);
```



# Home Assignment

- Conduct a desktop research and learn more about file descriptor



# Thank you