#### System Calls 2

**Aim:**

Write programs for illustrating the working of the following system calls used with directories and files in Linux :

**opendir(), readdir(), open(), read(), stat(), write(), close()**

**Description:**

**opendir():** This system call is used to open a directory stream corresponding to the directory name specified. It returns a pointer to a directory stream that can be used in subsequent calls to readdir() to read the contents of the directory.

**readdir()**: This system call is used to read the next directory entry from the directory stream opened by opendir(). It returns a pointer to a structure containing information about the next directory entry, such as the filename and file attributes. It returns NULL when the end of the directory is reached or if an error occurs.

**open()**: This system call is used to open a file or create a new file if it does not exist. It takes a filename and a set of flags as arguments to specify the mode of opening (read, write, or both) and other options. It returns a file descriptor, which is an integer representing the file in subsequent I/O operations.

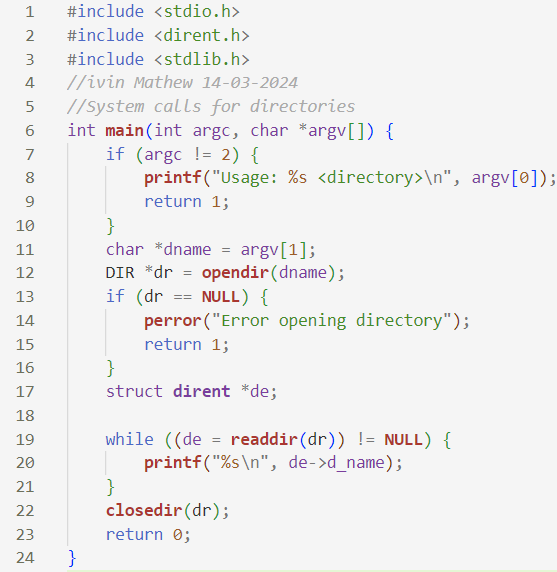
**read()**: This system call is used to read data from an open file descriptor into a buffer. It takes the file descriptor, a buffer where the data will be stored, and the number of bytes to read as arguments. It returns the number of bytes read, or 0 if the end of the file has been reached, or -1 if an error occurs.

**stat():** This system call is used to retrieve information about a file, such as its size, permissions, and timestamps. It takes the filename as an argument and fills a structure with information about the file. The stat() system call is often used in conjunction with open() and read() to perform file operations.

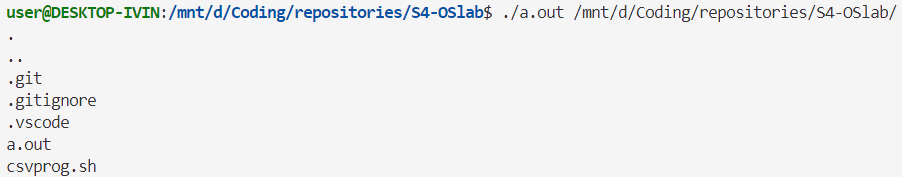
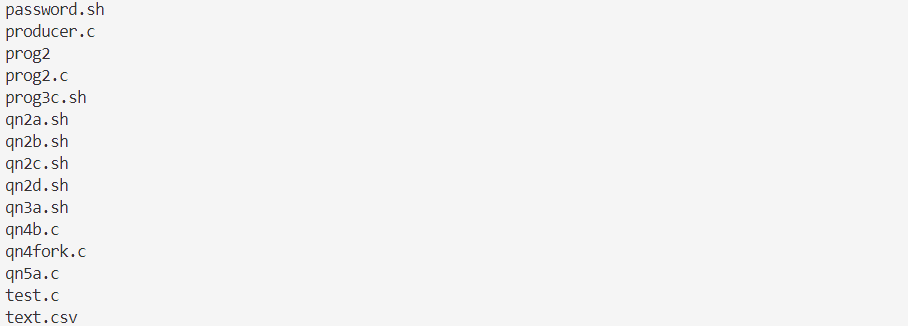
**write()**: This system call is used to write data from a buffer to an open file descriptor. It takes the file descriptor, a buffer containing the data to be written, and the number of bytes to write as arguments. It returns the number of bytes written, or -1 if an error occurs.

**close()**: This system call is used to close an open file descriptor. It takes the file descriptor as an argument and releases any resources associated with it. After a file descriptor is closed, it cannot be used for further I/O operations.

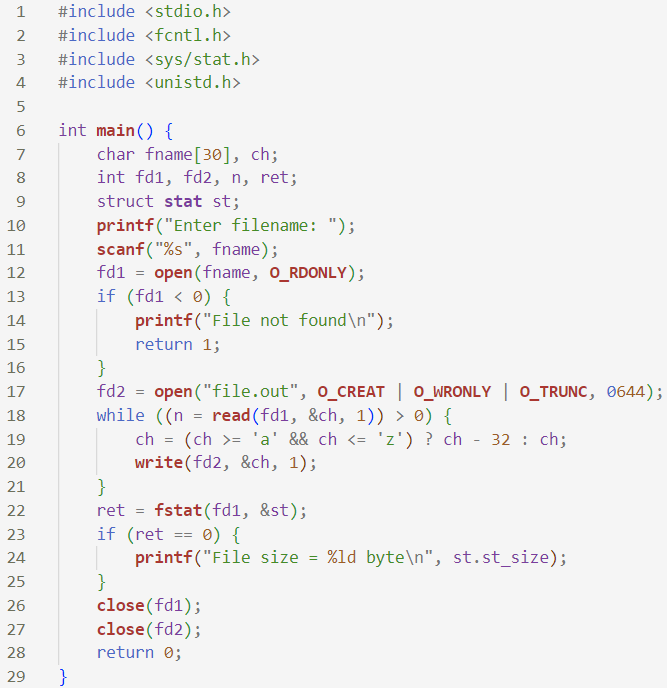
**Program 1**



**Sample input and output**

**** ****

**Program 2**

****

**Sample input and output**



**Result**

The programs have been executed and output has been verified.