BAHIRDAR UNIVERSITY

Department of Software Engineering

"Operating System and System Programming"

Individual Assignment "Bodhi OS Documentation"

Name: Eyuel Yeshambel Gebeyehu

ID No: BDU1601474

Section: B

Submitted To: Wendimu B.

Submission Date: 16/08/2017 E.C

System Call Implementation in Linux (C++)

The system **fstat()** – Retrieves file status information using a file descriptor (similar to stat(), but works with open file descriptors).

The fstat() system call in Bodhi Linux, a lightweight Linux distribution based on Ubuntu, is used to retrieve metadata about an open file specified by a file descriptor. It is part of the POSIX-compliant system calls available in Linux, including Bodhi OS, and is declared in the <sys/stat.h> header.

Key Points about fstat():

- **Purpose**: Obtains file status information, such as file size, permissions, ownership, timestamps (e.g., last access, modification), inode number, and file type, storing it in a struct stat buffer.
- **Usage in Bodhi OS**: Since Bodhi Linux uses a Linux kernel and standard libraries like glibc, fstat() behaves as it does in other Linux systems. It's commonly used in applications to check file properties without needing the file's path, unlike stat() or lstat().
- **Relation to Bodhi OS**: Bodhi Linux, with its Moksha desktop and Ubuntu base, supports fstat() as part of its standard system call interface. It's used in file management tasks, such as by the Thunar file manager or other system utilities.

Code

```
#include <iostream>
#include <sys/stat.h>
 #include <fcntl.h>
 #include <unistd.h>
 using namespace std;
int main() {
    const char* filename = "example.txt";  // Replace with your
    file int fd = open(filename, O_RDONLY);

    if (fd == -1) {
        cout << "Error: Failed to open file '" << filename << "'" << endl;
        return 1;
    }
    struct stat file_info;
    if (fstat(fd, &file_info) == -1) {
```

```
cout << "Error: System call failed to retrieve file information" << endl;
    close(fd);
        return 1;
    }

    cout << "File Type & Permissions: " << file_info.st_mode << endl;
    cout << "Inode Number: " << file_info.st_ino << endl;
  cout << "Device ID: " << file_info.st_dev << endl;
    cout << "Number of Hard Links:"<<file_info.st_nlink<<endl;
    cout << "User ID of Owner: "<<file_info.st_uid<<endl;
    cout << "Group ID of Owner: "<<file_info.st_gid<<endl;
    cout << "File Size (bytes): " << file_info.st_size << endl;
    cout << "Last Access Time:"<<file_info.st_atime<<endl;
    cout << "Last Modification Time: " << file_info.st_mtime << endl;

    close(fd);
    return 0;
}
```