Practical-2

Name: Pranav Deshmukh
Class: 2nd year CSE
Roll No: B4-51
Subject: OS LAB

Right Time

CODE:

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

int main() {
    int choice;
    char filename[1024], ch, pattern[1024];
    struct stat fileStat;

    while (1) {
        printf("\nChoose an option:\n");
        printf("1. Create a file\n");
        printf("2. Read contents of a file\n");
        printf("3. Write to a file\n");
        printf("4. Read contents of a file in reverse order\n");
        printf("5. Search for a pattern in the file (like grep)\n");
        printf("6. Delete a file\n");
        printf("7. Print file status using stat\n");
        printf("8. Print file status using fstat\n");
        printf("9. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: {
                printf("Enter the filename to create: ");
                scanf("%s", filename);
                int fd_create = open(filename, O_CREAT | O_WRONLY, 0644);
```

```
    if (fd_create == -1) {
        perror("Error creating file");
    } else {
        printf("File '%s' created successfully.\n", filename);
        close(fd_create);
    }
    break;
}
case 2: {
    printf("Enter the filename to read: ");
    scanf("%s", filename);
    int fd_read = open(filename, O_RDONLY);
    if (fd_read == -1) {
        perror("Error opening file for reading");
    } else {
        printf("File contents:\n");
        while (read(fd_read, &ch, 1) > 0) {
            putchar(ch);
        }
        close(fd_read);
    }
    break;
}
case 3: {
    printf("Enter the filename to write to: ");
    scanf("%s", filename);
    printf("Enter the text to write to the file: ");
    getchar();
    fgets(pattern, sizeof(pattern), stdin);
    int fd_write = open(filename, O_WRONLY | O_APPEND);
    if (fd_write == -1) {
        perror("Error opening file for writing");
    } else {
        write(fd_write, pattern, strlen(pattern));
        printf("Successfully written to file.\n");
        close(fd_write);
    }
    break;
}
case 4: {
    printf("Enter the filename to read in reverse: ");
    scanf("%s", filename);
    int fd_reverse = open(filename, O_RDONLY);
    if (fd_reverse == -1) {
        perror("Error opening file for reading");
```

```
    } else {
        off_t file_size = lseek(fd_reverse, 0, SEEK_END);
        for (off_t i = 1; i <= file_size; i++) {
            lseek(fd_reverse, -i, SEEK_END);
            read(fd_reverse, &ch, 1);
            putchar(ch);
        }
        close(fd_reverse);
    }
    break;
}
case 5: {
    printf("Enter the filename to search: ");
    scanf("%s", filename);
    printf("Enter the pattern to search for: ");
    getchar();
    fgets(pattern, sizeof(pattern), stdin);
    pattern[strcspn(pattern, "\n")] = 0;
    int fd_grep = open(filename, O_RDONLY);
    if (fd_grep == -1) {
        perror("Error opening file for searching");
    } else {
        int found = 0;
        char buffer[1024];
        int bytes_read;
        while ((bytes_read = read(fd_grep, buffer, sizeof(buffer))) > 0) {
            if (strstr(buffer, pattern) != NULL) {
                found = 1;
                break;
            }
        }
        if (found) {
            printf("Pattern '%s' found in the file.\n", pattern);
        } else {
            printf("Pattern '%s' not found in the file.\n", pattern);
        }
        close(fd_grep);
    }
    break;
}
case 6: {
    printf("Enter the filename to delete: ");
    scanf("%s", filename);
    if (remove(filename) == 0) {
        printf("File '%s' deleted successfully.\n", filename);
```

```
            } else {
                perror("Error deleting file");
            }
            break;
        }
        case 7: {
            printf("Enter the filename to check status using stat: ");
            scanf("%s", filename);
            if (stat(filename, &fileStat) == -1) {
                perror("Error retrieving file status");
            } else {
                printf("File Size: %ld bytes\n", fileStat.st_size);
                printf("File Permissions: %o\n", fileStat.st_mode & 0777);
                printf("Last Modified Time: %ld\n", fileStat.st_mtime);
            }
            break;
        }
        case 8: {
            printf("Enter the filename to check status using fstat: ");
            scanf("%s", filename);
            int fd_fstat = open(filename, O_RDONLY);
            if (fd_fstat == -1) {
                perror("Error opening file");
            } else {
                if (fstat(fd_fstat, &fileStat) == -1) {
                    perror("Error retrieving file status");
                } else {
                    printf("File Size: %ld bytes\n", fileStat.st_size);
                    printf("File Permissions: %o\n", fileStat.st_mode & 0777);
                    printf("Last Modified Time: %ld\n", fileStat.st_mtime);
                }
                close(fd_fstat);
            }
            break;
        }
        case 9:
            printf("Exiting program...\n");
            exit(0);
        default:
            printf("Invalid choice. Please try again.\n");
      }
   }
   return 0;
}
```

OUTPUT:

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 1
Enter the filename to create: 51
File '51' created successfully.
```

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 2
Enter the filename to read: 51
File contents:
What is so special in this file
```

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 3
Enter the filename to write to: 51
Enter the text to write to the file: What is so special in this file
Successfully written to file.
```

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 4
Enter the filename to read in reverse: 51
```

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 5
Enter the filename to search: 51
Enter the pattern to search for: grep
Pattern 'grep' not found in the file.
```

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 7
Enter the filename to check status using stat: 51
File Size: 32 bytes
File Permissions: 644
Last Modified Time: 1738742514
```

```
Choose an option:
1. Create a file
2. Read contents of a file
3. Write to a file
4. Read contents of a file in reverse order
5. Search for a pattern in the file (like grep)
6. Delete a file
7. Print file status using stat
8. Print file status using fstat
9. Exit
Enter your choice: 6
Enter the filename to delete: 51
File '51' deleted successfully.
```