

B.BHANUTEJA REDDY-192325016

24.Design a C program to demonstrate UNIX system calls for file management.

AIM:

To design a C program that demonstrates UNIX system calls for file management.

ALGORITHM:

1. Use open() to create and open a file.
2. Use write() to write data to the file.
3. Use lseek() to adjust the file pointer.
4. Use read() to read data from the file.
5. Use close() to close the file.

CODE:

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

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

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

```
#include <string.h>
```

```
int main() {
```

```
    char buffer[100];
```

```
    int file = open("example.txt", O_CREAT | O_RDWR, 0644);
```

```
    if (file < 0) {
```

```
        perror("Error opening file");
```

```
        return 1;
```

```
    }
```

```
write(file, "Hello, UNIX system calls!", 25);

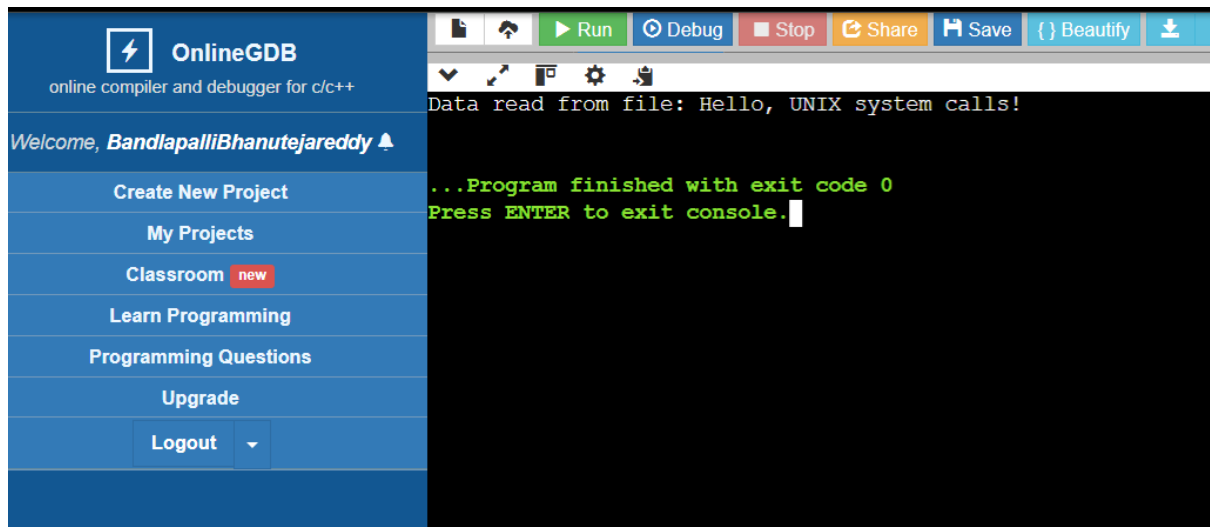
lseek(file, 0, SEEK_SET);

int bytesRead = read(file, buffer, sizeof(buffer) - 1);
if (bytesRead > 0) {
    buffer[bytesRead] = '\0';
    printf("Data read from file: %s\n", buffer);
}

close(file);

return 0;
}
```

OUTPUT:

The screenshot shows the OnlineGDB web interface. On the left is a blue sidebar with the OnlineGDB logo and navigation links: 'Create New Project', 'My Projects', 'Classroom' (with a 'new' badge), 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area has a top toolbar with buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and a download icon. Below the toolbar is a terminal window with a black background and green text. The terminal output shows 'Data read from file: Hello, UNIX system calls!' followed by '...Program finished with exit code 0' and 'Press ENTER to exit console.' with a cursor at the end.