

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

Aim

To write a C program that demonstrates the use of UNIX system calls for file management, including file creation, opening, reading, writing, and closing.

Algorithm

1. Start the program.
2. Use `creat()` or `open()` to create or open a file.
3. Write data to the file using `write()`.
4. Close the file using `close()`.
5. Reopen the file using `open()` in read mode.
6. Read data from the file using `read()`.
7. Display the read data on the console.
8. Close the file.
9. End the program.

Procedure

1. Import necessary headers (like `fcntl.h` and `unistd.h`).
2. Define the file name and data to write.
3. Use `creat()` or `open()` to create/open a file.
4. Use `write()` to write data into the file.
5. Close the file using `close()`.
6. Use `open()` to reopen the file in read mode.
7. Use `read()` to read the data from the file into a buffer.
8. Display the content read from the file.
9. Close the file using `close()`.

Code:

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

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

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

```
int main() {
```

```
int fd;

char buffer[100];

const char *data = "Hello, UNIX file management!";


fd = creat("example.txt", 0644);

write(fd, data, 27);

close(fd);


fd = open("example.txt", O_RDONLY);

read(fd, buffer, 27);

buffer[27] = '\0';

printf("Read data: %s\n", buffer);

close(fd);


return 0;

}
```

Result

The program demonstrates the creation, writing, reading, and closing of a file using UNIX system calls. When executed, the content of the file ("Hello, UNIX file management!") is read and displayed on the console.

Output:

```
Read data: Hello, UNIX file management
```

```
...Program finished with exit code 0  
Press ENTER to exit console. 
```