System call in beos operating system

Unmount()

Implementation of system calls.

A **system call** is a way for user programs to request services or actions from the operating system's kernel. For example, reading files, mounting/unmounting disks, or managing processes.

Objective

The objective of this guide is to demonstrate the process of writing, saving, and compiling a simple C/C++ program in the **BeOS operating system** using the **Terminal**. This includes:

Launching and using **StyledEdit**, the default BeOS text editor, to write source code.

Understanding the basic structure of a C/C++ program.

Compiling the source code using the **g++ compiler** from the Terminal

Verifying the successful creation and execution of the compiled program.

This hands-on process will help users get familiar with coding and compiling in the BeOS environment, which is especially useful for learning low-level system programming and understanding how classic UNIX-like operating systems handle development workflows.

1. Open BeOS Terminal

Go to **Applications > Terminal**

and then I entered the following command

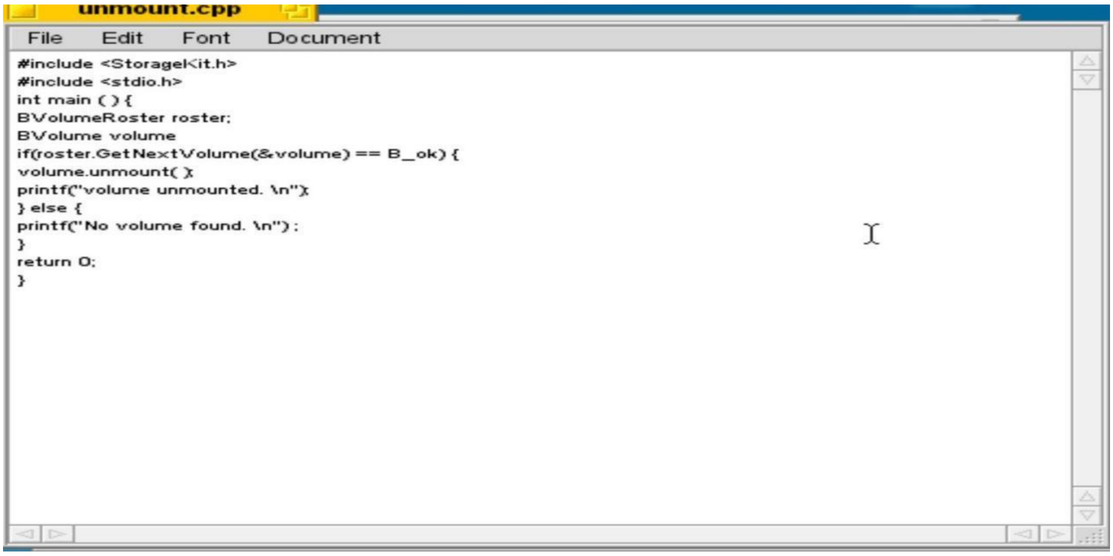
styledEdit

From here, you'll be able to write, compile, and run your code.

Once you're here, this is where you'll be doing all your coding—writing, compiling, and running your programs

2. Create the Source Code File of unmount() (this is my sytem call)

Open **StyledEdit** (BeOS text editor) on this way like below screen shoot



```
#include <StorageKit.h>
#include <stdio.h>
int main ( ) {
BVolumeRoster roster;
BVolume volume
if(roster.GetNextVolume(&volume) == B_ok) {
volume.unmount( );
printf("volume unmounted. \n");
} else {
printf("No volume found. \n") ;
}
return 0;
}
```

Save the file in this location:
/boot/home/mount.cpp

3 Compile the Code

Go to Terminal and type the following command:

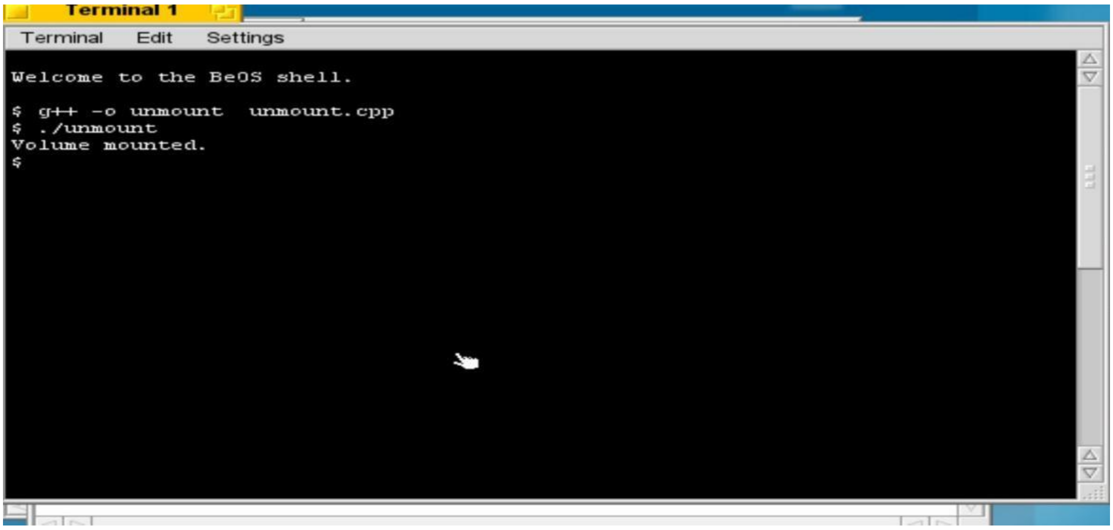
```
g++ -o mount mount.cpp
```

4. Run the Program

Still in Terminal, run: on command

```
./mount
```

On this way and show that wanted output



```
Welcome to the BeOS shell.

$ g++ -o unmount  unmount.cpp
$ ./unmount
Volume mounted.
$
```

Successfully wrote and compiled a basic program in BeOS —

everythiing's up and running smoothly!