

## 4, implement system call

I will implement system call chroot() - Changes the root directory for a process.

The chroot() system call in Unix-like operating systems, including Ubuntu 20.04, is used to change the root directory of the current running process and its children to a specified directory. This effectively creates a new root filesystem for the process, isolating it from the rest of the filesystem hierarchy.

Steps to implement

Sudo mkdir /newroot --> to create the newroot directory

Sudo cp bin/bash/ newroot/ --> to copy bash into /newroot

Ldd/bin/bash --> to list bash dependencies

Sudo mkdir -p /newroot/lib/x86\_64-linux-gnu --> create necessary directories

sudo cp /lib/x86\_64-linux-gnu/libc.so.6 /newroot/lib/x86\_64-linux-gnu/ --> to copy the dependencies.

nano chroot.c --> to go to text editor

Paste the correct code which i will show in photo

gcc chroot.c -o chroot --> to compile

Sudo ./chroot/newroot --> to run the program

Let's show in photo which i took from my computer

A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background with light-colored text. It displays the following command-line session:

```
yordaos@yordaos:~$ sudo mkdir [sudo] password for yordaos: mkdir: missing operand Try 'mkdir --help' for more information.
yordaos@yordaos:~$ sudo mkdir /new root
yordaos@yordaos:~$ sudo mkdir /newroot
yordaos@yordaos:~$ sudo cp /bin/bash /newroot
yordaos@yordaos:~$ ldd /bin/bash
    linux-vdso.so.1 (0x00007fff733de000)
    libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007f2391374000)
    libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f239136e000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f239117c000)
    /lib64/ld-linux-x86_64.so.2 (0x00007f23914e4000)
yordaos@yordaos:~$ sudo mkdir -p /newroot/lib/x86_64-linux-gnu
yordaos@yordaos:~$ sudo cp /lib/x86_64-linux-gnu/libtinfo.so.6 /newroot/lib/x86_64-linux-gnu/
yordaos@yordaos:~$ sudo cp /lib/x86_64-linux-gnu/libdl.so.2 /newroot/lib/x86_64-linux-gnu/
yordaos@yordaos:~$
```

```
GNU nano 4.8          chroot.c          Modified
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>

int main(int argc, char *argv[]){
if (argc !=2){
fprintf(stderr, "usage: %s<new_root_directory>\n", argv[0]);
return 1;
}
const char *new_root=argv[1];
if (chroot(new_root)!=0){
fprintf(stderr, "chroot() failed: %s\n", strerror(errno));
}
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify  
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell

```
GNU nano 4.8          chroot.c          Modified
const char *new_root=argv[1];
if (chroot(new_root)!=0){
fprintf(stderr, "chroot() failed: %s\n", strerror(errno));
return 1;
}
if(chdir("/")!=0){
fprintf(stderr, "chdir() failed: %s\n", strerror(errno));
return 1;
}
char *command[]={"/bin/bash", Null};
execv(command[0], command);
fprintf(stderr, "execv() failed: %s\n", strerror(errno));
return 1;
}

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell
```

```
collect2: error: ld returned 1 exit status
yordaos@yordaos:~$ nano chroot.c
yordaos@yordaos:~$ nano chroot1.c
yordaos@yordaos:~$ gcc chroot1.c
chroot1.c:1:2: error: invalid preprocessing directive #define_G
NU_SOURCE
  1 | #define_GNU_SOURCE
   | ^
yordaos@yordaos:~$ nano chroot1.c
yordaos@yordaos:~$ gcc chroot1.c
yordaos@yordaos:~$ sudo ./chroot1 /home/yordaos/new_root
[sudo] password for yordaos:
sudo: ./chroot1: command not found
yordaos@yordaos:~$ sudo cp /bin/bash/newroot/
[sudo] password for yordaos:
cp: missing destination file operand after '/bin/bash/newroot/'
Try 'cp --help' for more information.
yordaos@yordaos:~$ sudo cp /bin/bash /newroot/
yordaos@yordaos:~$
```