

1. Use online editor for shell environment
<https://bellard.org/jslinux/vm.html?cpu=riscv64&url=fedora33-riscv-xwin.cfg&graphic=1&mem=256>
2. Alternatively install virtual dosbox in your systems

Lab pgm 9a :

a) Shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, outputs the common permissions, otherwise outputs each file name followed by its permissions.

```

if [ $# != 2 ]
then
echo "Invalid input!!!"
else
p1=`ls -l $1|cut -d " " -f1`
p2=`ls -l $2|cut -d " " -f1`
if [ $p1 == $p2 ]
then
echo "the file permissions are same and it is : "
echo "$p1"
else
echo "The file permissions are different"
echo "$1 : $p1"
echo "$2 : $p2"
fi
fi

```

OUTPUT

command to create the file vi pgm9.sh

use ls -l command to get list of files and their permission

Use chmod 777 pgmname to change permissions

command to execute ./9a.sh filename1 filename2

The screenshot shows a terminal window titled '@localhost:~'. The command 'ls -l' is run, displaying a list of files with their permissions, sizes, and modification dates. The output includes:

```
[root@localhost ~]# ls -l
total 24
-rw----- 1 1000 root 277 May 1 12:36 9a.sh
-rw-r--r-- 1 root root 114 Dec 26 15:49 bench.py
-rw-r--r-- 1 root root 185 Sep 9 2018 hello.c
-rw-r--r-- 1 root root 0 May 1 12:33 n.sh
-rwxrwxrwx 1 root root 78 May 1 12:17 p2.sh
-rw-r--r-- 1 root root 228 May 1 12:00 pgm
-rw-r--r-- 1 root root 0 May 1 11:48 pgm9
---x---x--- 1 root root 227 May 1 12:31 pgm9.sh
[root@localhost ~]# sh 9a.sh bench.py n.sh
the file permissions are same and it is :
-rw-r--r--
```

Pgm 9b :

C program that creates a child process to read commands from the standard input and execute them

```
#include<stdio.h>
int main()
{
    int ch,rv;
    char cmd[10];
    rv=fork();
    if(rv==0)
    {
        do
        {
            printf("\nEnter a command\n");
            scanf("%s",cmd);
            system(cmd);
            printf("\n1 : continue\n0 : exit\n");
            scanf("%d",&ch);
        }
        while(ch!=0);
    }
    else
    {
        wait(0);
        printf("\nChild terminated\n");
    }
    return 0;
}
```

output

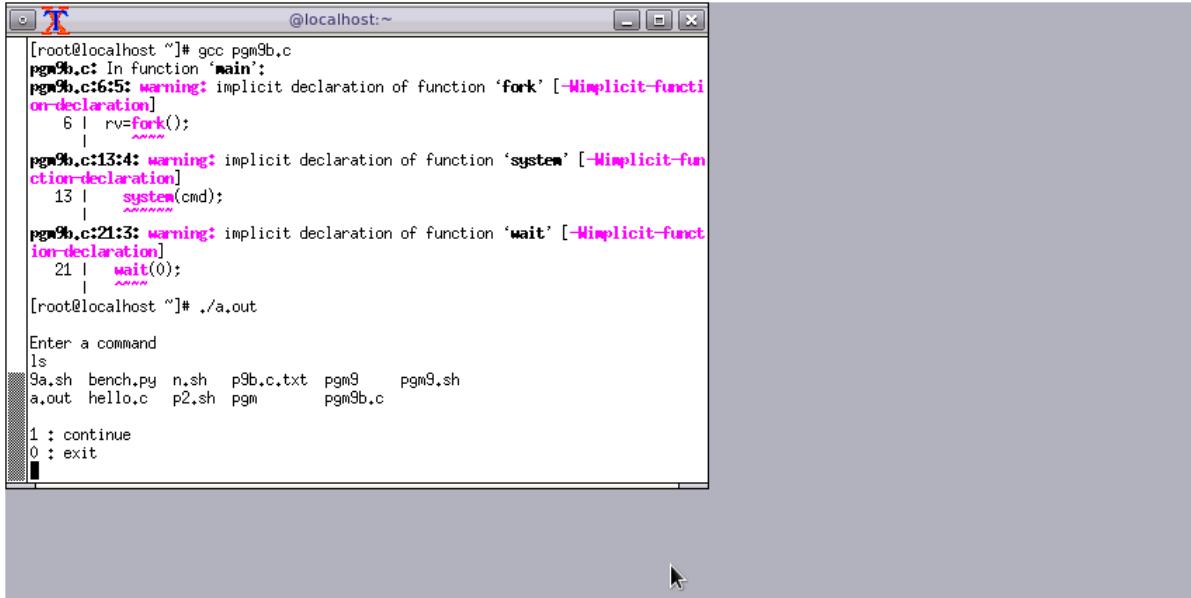
use the same online editor.

Command to write pgm in online editor : vi pgm9b.c or u can upload file directly

command to execute c file(C compiler) : gcc pgm9b.c

clear error if any, warnings are ok

Command to run the code : ./a.out



The screenshot shows a terminal window titled 'T' with the command '@localhost:~'. Inside the window, the user runs 'gcc pgm9b.c' which produces several warning messages about implicit declarations of 'fork', 'system', and 'wait'. The user then runs './a.out' and is prompted to enter a command. They type 'ls' and see a list of files including '9a.sh', 'bench.py', 'n.sh', 'p9b.c.txt', 'pgm9', 'pgm9.sh', 'a.out', 'hello.c', 'p2.sh', 'pgm', and 'pgm9b.c'. At the bottom, there is a prompt for continuation with '1 : continue' and '0 : exit'.

```
[root@localhost ~]# gcc pgm9b.c
pgm9b.c: In function ‘main’:
pgm9b.c:6:5: warning: implicit declaration of function ‘fork’ [-Wimplicit-function-declaration]
  6 |   rv=fork();
     |
pgm9b.c:13:4: warning: implicit declaration of function ‘system’ [-Wimplicit-function-declaration]
  13 |   system(cmd);
     |
pgm9b.c:21:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-function-declaration]
  21 |   wait(0);
     |
[root@localhost ~]# ./a.out
Enter a command
ls
9a.sh  bench.py  n.sh  p9b.c.txt  pgm9    pgm9.sh
a.out  hello.c   p2.sh  pgm      pgm9b.c
1 : continue
0 : exit
```