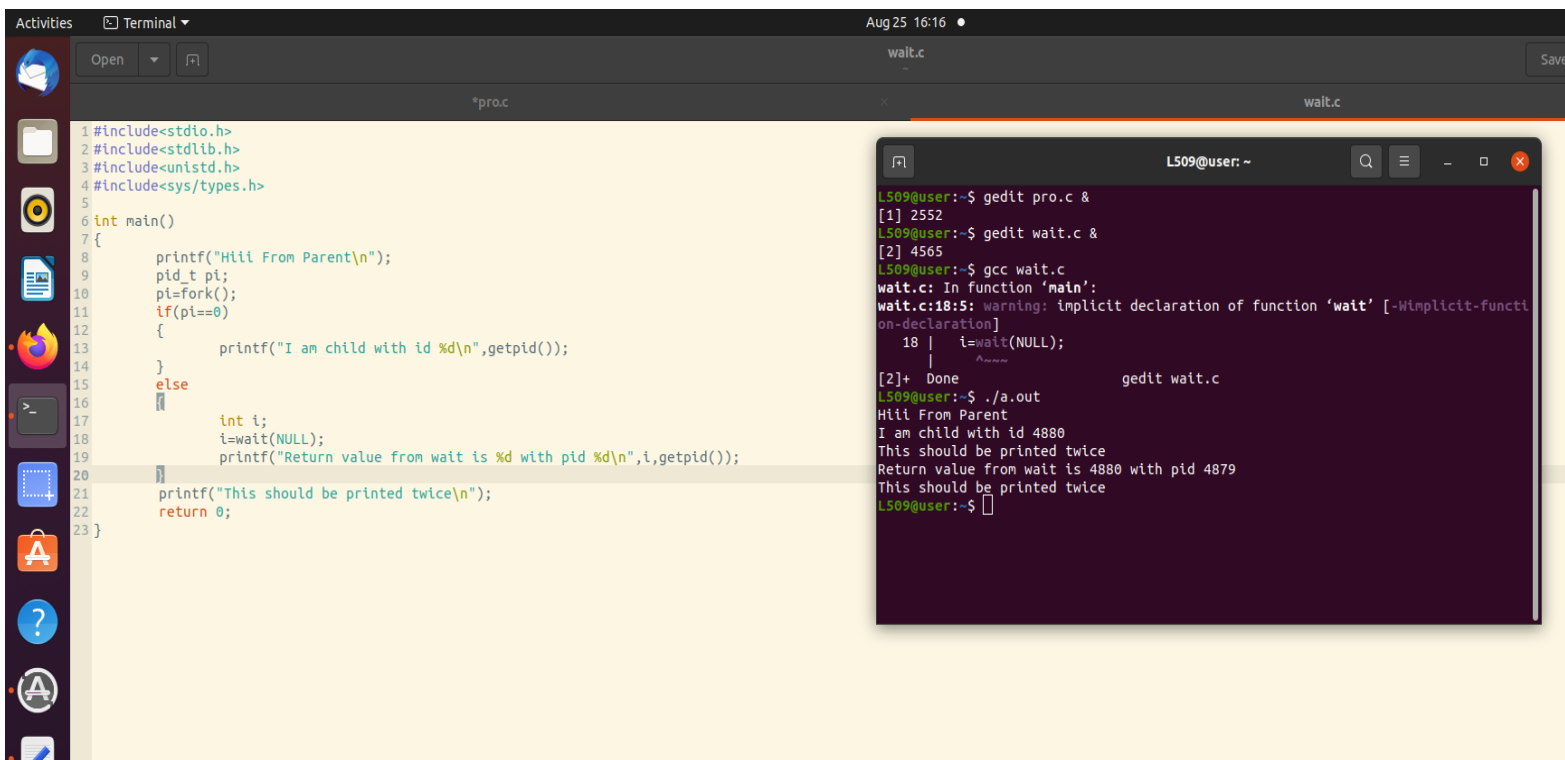



## SWARAJ(CS21B2024) OS LAB 4 PROGRAM 3



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<sys/types.h>
5
6 int main()
7 {
8     printf("Hi! From Parent\n");
9     pid_t pi;
10    pi=fork();
11    if(pi==0)
12    {
13        printf("I am child with id %d\n",getpid());
14    }
15    else
16    {
17        int i;
18        i=wait(NULL);
19        printf("Return value from wait is %d with pid %d\n",i,getpid());
20    }
21    printf("This should be printed twice\n");
22    return 0;
23 }
```

```
L509@user: ~
L509@user:~$ gedit pro.c &
[1] 2552
L509@user:~$ gedit wait.c &
[2] 4565
L509@user:~$ gcc wait.c
wait.c: In function 'main':
wait.c:18:5: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
    18 |     i=wait(NULL);
        |         ^
[2]+  Done                  gedit wait.c
L509@user:~$ ./a.out
Hi! From Parent
I am child with id 4880
This should be printed twice
Return value from wait is 4880 with pid 4879
This should be printed twice
L509@user:~$
```

WAIT()  
WAIT()

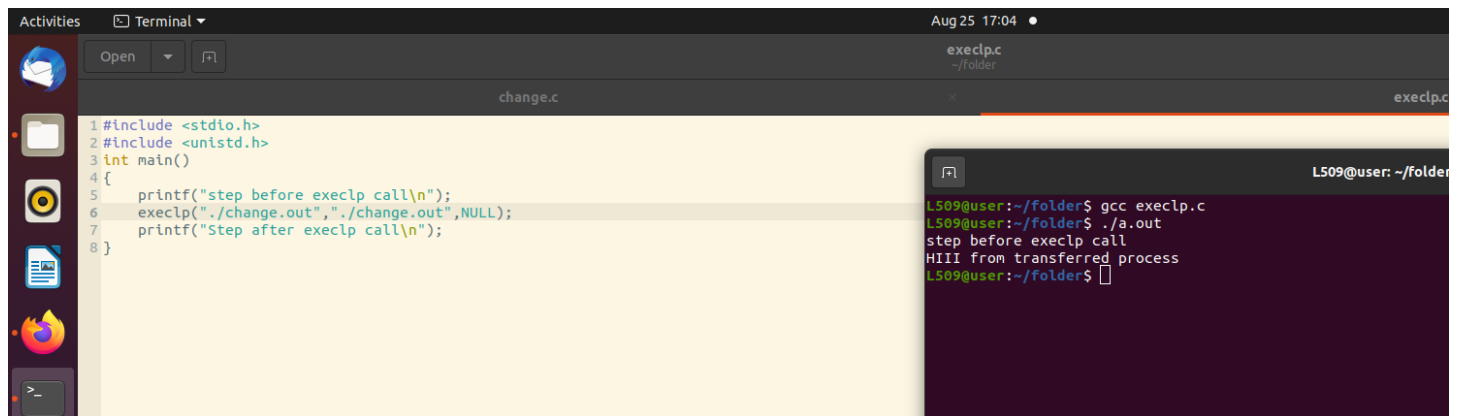


```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<sys/types.h>
5
6 int main()
7 {
8     printf("Hi! From Parent\n");
9     pid_t pi1,pi2;
10    pi1=fork();
11    if(pi1==0)
12    {
13        pi2=fork();
14        if(pi2==0)
15        {printf("I am child of child with id %d and parent id as %d\n",getpid(),getppid());}
16        else
17        {printf("I am child with id %d and parent id as %d\n",getpid(),getppid());}
18    }
19    else
20    {
21        int i;
22        i=waitpid(pi1);
23        printf("Return value from waitpid is %d with pid %d\n",i,getpid());
24    }
25
26    return 0;
27 }
```

```
L509@user: ~
waitpid.c: In function 'main':
waitpid.c:23:5: warning: implicit declaration of function 'waitpid' [-Wimplicit-function-declaration]
    23 |     i=waitpid(pi2);
        |         ^
L509@user:~$ ./a.out
Hi! From Parent
Return value from waitpid is -1 with pid 5309
I am child with id 5310 and parent id as 1419
I am child of child with id 5311 and parent id as 5310
L509@user:~$ gcc waitpid.c
waitpid.c: In function 'main':
waitpid.c:22:5: warning: implicit declaration of function 'waitpid' [-Wimplicit-function-declaration]
    22 |     i=waitpid(pi1);
        |         ^
L509@user:~$ ./a.out
Hi! From Parent
I am child with id 5365 and parent id as 5364
I am child of child with id 5366 and parent id as 5365
Return value from waitpid is 5365 with pid 5364
L509@user:~$
```

WAIT PID()

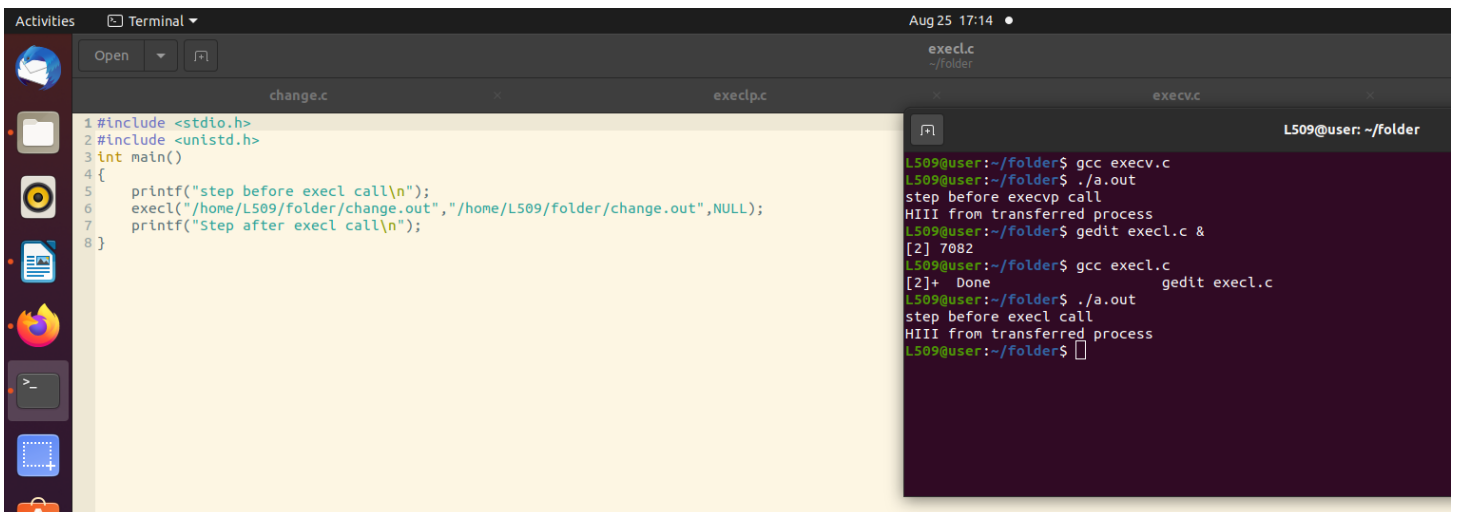
## SWARAJ(CS21B2024) OS LAB 4 PROGRAM 3



```
1#include <stdio.h>
2#include <unistd.h>
3int main()
4{
5    printf("step before execlp call\n");
6    execlp("./change.out", "./change.out", NULL);
7    printf("Step after execlp call\n");
8}
```

```
L509@user: ~/folder
L509@user:~/folder$ gcc execlp.c
L509@user:~/folder$ ./a.out
step before execlp call
HIII from transferred process
L509@user:~/folder$
```

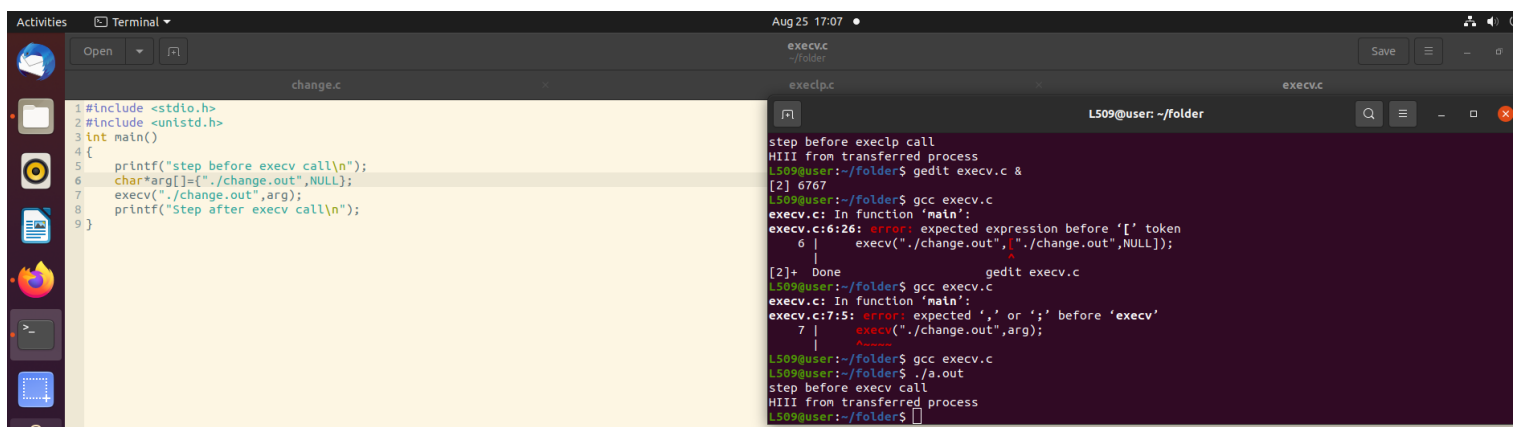
EXECLP()



```
1#include <stdio.h>
2#include <unistd.h>
3int main()
4{
5    printf("step before execl call\n");
6    execl("/home/L509/folder/change.out", "/home/L509/folder/change.out", NULL);
7    printf("Step after execl call\n");
8}
```

```
L509@user: ~/folder
L509@user:~/folder$ gcc execl.c
L509@user:~/folder$ ./a.out
step before execl call
HIII from transferred process
L509@user:~/folder$ gedit execl.c &
[2] 7082
L509@user:~/folder$ gcc execl.c
[2]+ Done gedit execl.c
L509@user:~/folder$ ./a.out
step before execl call
HIII from transferred process
L509@user:~/folder$
```

EXECL()

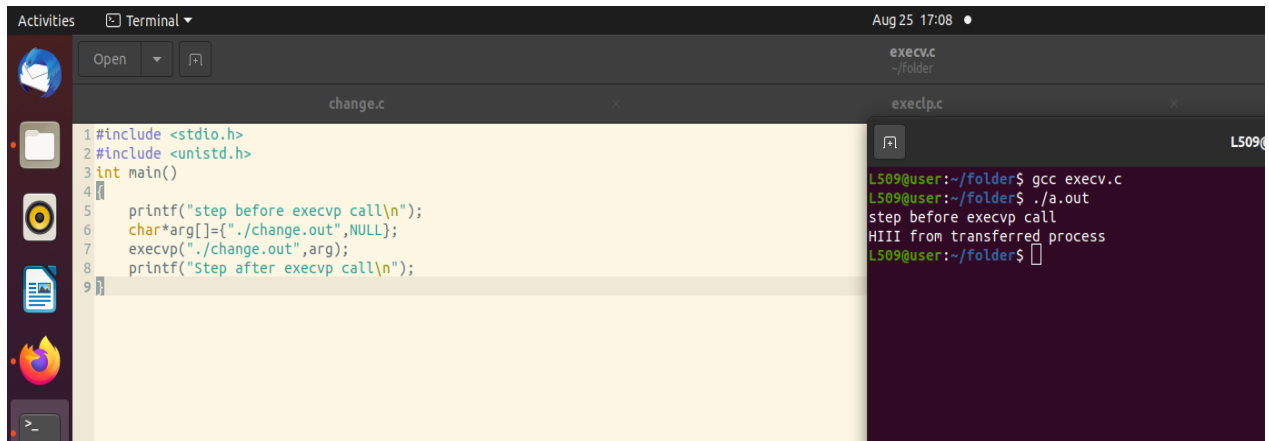


```
1#include <stdio.h>
2#include <unistd.h>
3int main()
4{
5    printf("step before execv call\n");
6    char*arg[]={"./change.out", NULL};
7    execv("./change.out", arg);
8    printf("Step after execv call\n");
9}
```

```
L509@user: ~/folder
L509@user:~/folder$ gcc execv.c
L509@user:~/folder$ ./a.out
step before execv call
HIII from transferred process
L509@user:~/folder$ gedit execv.c &
[2] 6767
L509@user:~/folder$ gcc execv.c
execv.c: In function 'main':
execv.c:6:26: error: expected expression before '[' token
6 |     execv("./change.out", ["./change.out", NULL]);
  |                          ^
[2]+ Done gedit execv.c
L509@user:~/folder$ gcc execv.c
execv.c: In function 'main':
execv.c:7:5: error: expected '(', or ';' before 'execv'
7 |     execv("./change.out", arg);
  |     ^~~~~
L509@user:~/folder$ gcc execv.c
L509@user:~/folder$ ./a.out
step before execv call
HIII from transferred process
L509@user:~/folder$
```

EXECV()

## SWARAJ(CS21B2024) OS LAB 4 PROGRAM 3



The screenshot shows a Linux desktop environment. On the left is a vertical dock with icons for Activities, Files, Applications, and a terminal. The main window is a code editor with two tabs: 'change.c' and 'execvp.c'. The 'change.c' tab is active, showing the following C code:

```
1#include <stdio.h>
2#include <unistd.h>
3int main()
4{
5    printf("step before execvp call\n");
6    char*arg[]={"/change.out",NULL};
7    execvp("./change.out",arg);
8    printf("Step after execvp call\n");
9}
```

To the right of the code editor is a terminal window. It shows the following commands and output:

```
L509@user:~/folder$ gcc execvp.c
L509@user:~/folder$ ./a.out
step before execvp call
HIII from transferred process
L509@user:~/folder$
```

EXECVP()

wait( ).

Sworaj

CS21B2024

wait( ) - wait for child process to end.

Suspends the calling process until any one of the child process ends.

wait(status\_ptr)

If there is no child process to wait for return value is 0. In case successful it returns pid of child.

waitpid( )

waitpid( pid\_t pid, int \* status\_ptr, int options)

Suspends calling till particular child process gets over.

exec( )

Call replaces the image of the current process with new process image specified by the path. Hence execution of current process gets transferred.

Swaraj CS21B2024

`execvp()` In this path need not referenced  
complicated but is same as `exec()`.

`execv()`

It has same use like `exec()` but  
syntax is different.

`execvp()` It has same use like `execvp()` but

syntax is different.