

ARQUITECTURA CLIENTE SERVIDOR

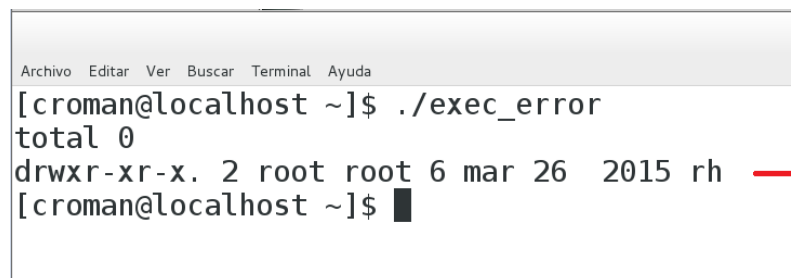
TAREA # 3

Tomando el siguiente código de ejemplo, al que llamaremos `exec_error.c`

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

int main(void)
{
    char *args[] = {"ls", "-l", "/opt", NULL};
    if(execvp("ls", args) == -1) {
        exit(EXIT_FAILURE);
    }
    puts("a partir de aqui no se ejecuta");
    exit(EXIT_SUCCESS);
}
```

Compíllalo y muestra aquí la captura de pantalla de la ejecución:



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[croman@localhost ~]$ ./exec_error
total 0
drwxr-xr-x. 2 root root 6 mar 26  2015 rh
[croman@localhost ~]$
```

Tu salida puede variar
con respecto al del
ejemplo

Ahora vamos a modificar el código para que genere error. Para ello, reemplazamos el `execvp` por `execve`, como lo indica la modificación que está en amarillo:

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

int main(void)
{
    char *args[] = {"ls", "-l", "/opt", NULL};
    if(execve("ls", args) == -1) {
        exit(EXIT_FAILURE);
    }
    puts("a partir de aqui no se ejecuta");
}
```

```
    exit(EXIT_SUCCESS);  
}
```

Al compilarlo, marca error de compilación. Indica aquí el error que te marca:

Pero el error que queremos generar es un error en tiempo de ejecución, por lo que hay que corregir este error en tiempo de compilación. Para ello falta agregar el tercer parámetro a la función `execve`, que es el vector de variables de entorno personalizadas. Como para este ejemplo no queremos variables de entorno personalizadas, entonces pasamos `NULL`:

Modifica el código, agregando lo que está en amarillo.

```
if(execve("ls", args, NULL) == -1)
```

Ahora ya compila. Ejecútalo y agrega aquí la captura de pantalla de la ejecución:

Aquí ya generó un error en tiempo de ejecución, pero no es visible en pantalla. Ahora investiga qué es la función `perror()`, para qué se utiliza y modifica el código, agregando lo que está en amarillo:

```
if(execve("ls", args, NULL) == -1) {  
    perror("execve");  
    exit(EXIT_FAILURE);  
}
```

Ahora compila esta versión. Ejecútalo y agrega aquí la captura de pantalla de la ejecución:

Explica por qué da ese mensaje de error:

Además de la función `perror()`, investiga la variable `errno` y las macros `EXIT_SUCCESS` y `EXIT_FAILURE`.