

# Llamadas exec

---

Las llamadas `exec` son una familia de funciones que nos permiten reemplazar el código del proceso actual por el código del programa que se pasa como parámetro.

## `execl`

---

Esta llamada toma el programa a ejecutar en una lista de cadenas terminada en `NULL`:

```
int execl(char *ruta_a_ejecutable, char *prog, ...);
```

Ejemplo de uso:

```
execl("/bin/ls", "ls", "-la", NULL);  
perror("fallo en exec");  
exit(EXIT_FAILURE);
```

## `execv`

---

Esta llamada toma el programa a ejecutar en un array de cadenas:

```
int execv(char *ruta_a_ejecutable, char *prog[]);
```

Ejemplo de uso:

```
char *prog[] = { "ls", "-la", NULL };  
  
execv("/bin/ls", prog);  
perror("fallo en exec");  
exit(EXIT_FAILURE);
```

## `execlp`

---

Similar a `execv` pero el primer parámetro se trata del nombre del programa y se confía en el contenido de la variable de entorno `$PATH`.

```
int execlp(char *nombre_ejecutable, char *prog, ...);
```

Ejemplo de uso:

```
execlp("ls", "ls", "-la", NULL);  
perror("fallo en exec");  
exit(EXIT_FAILURE);
```

## `execvp`

---

Similar a `execv` pero el primer parámetro se trata del nombre del programa y se confía en el contenido de la variable de entorno `$PATH`.

```
int execvp(char *nombre_ejecutable, char *prog[]);
```

Ejemplo de uso:

```
char *prog[] = { "ls", "-la", NULL };  
  
execvp("ls", prog);  
perror("fallo en exec");  
exit(EXIT_FAILURE);
```