

# Chiamata di Sistema: Exec

Leonardo Bizzoni

January 26, 2024

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

int main(void) {
    printf("Elenco file in wd: \n");
    execl("/usr/bin/ls", "ls", "-ahl", (char *)NULL);
    /* execlp("ls", "ls", "-ahl", (char *)NULL); // cerca "ls" nel PATH */
    printf("Fine elenco.\n");
}
```

L'ultima chiamata printf() non viene effettuata in quanto execl() sostituisce l'immagine del processo corrente con una nuova immagine dedicata al programma dato in input alla funzione execl().

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main(void) {
    printf("Elenco file in wd: \n");

    int cpid = fork();
    if (cpid == 0) {
        execl("/usr/bin/ls", "ls", "-alh", (char *)NULL);
        perror("[execlp]"); // execlp on success termina il programma e on failure
                           // returns a value
    } else if (cpid > 0) {
        int wchild_pid = wait(NULL);

        if (wchild_pid < 0) {
```

```
    perror("[wait error]");
} else {
    printf("Fine elenco.\n");
}
}
}
```

---

In un programma multithreaded effettuare una chiamata *exec()* solitamente **termina tutti** i thread del processo in esecuzione.