Anotações - Guião 3

Data: 16/03/2023
Tags: #SO #C #uni #SoftwareEngineering

---

execl

`execl(path, nome do executável: *arg0, arg1, arg2, ..., argN, NULL)`
-> Apenas retorna valor em caso de erro, pois caso execute o código, é feita uma "substituição" do seu código pelo código do novo programa, não havendo retorno.

execlp

Semelhante à chamada anterior mas recorre aos caminhos registados na variável de ambiente *PATH* (ex.: apenas `ls` em vez de `/bin/ls`).

execv e execvp

`execv(const char *path, char *const argv[])`
(também necessita de NULL no final e o primeiro arg tem de ser o nome do executável)

> Exemplo de utilização do execvp

```c
#define MAX_ARGS 10

int main (int argc, char const *argv[]) {

    char **sep = malloc(sizeof(char *) * MAX_ARGS);
    memset(sep, 0, sizeof(sep));

    char *string = strdup("ls -l");
    char *aux = string;

    for (int i = 0; aux != NULL && i < MAX_ARGS; i++) {
        char *part = strsep(&aux, " ");
        int length = strlen(part);
        sep[i] = malloc(sizeof(char) * length);
        strncpy(sep[i], part, length);
    }

    int fres = fork();
    if (fres == 0) {
        int res = execvp(sep[0], sep);
        printf("Did not execute command (%d).\n", res);
        _exit(1);
    } else {
        int status;
        waitpid(fres, &status, 0);
        if (WIFEXITED(status)) {
            printf("Returned: %d\n", WEXITSTATUS(status));
        } else {
            printf("Error.\n");
        }
    }

    for (int i = 0; i < MAX_ARGS; i++) {
        free(sep[i]);
    }
```

```
    free(sep);

    return 0;
}
```