RELATORIO DE SALA DE AULA 2

UNIEVANGELICA

SISTEMA OPERACIONAIS

GABRIEL MENDONÇA

VICTOR HUGO PIGNATA

LUCAS NUNES PINTO

Na segunda aula iniciamos com uma breve explicação sobre a funções exec que constituem na verdade uma família de funções e são conhecidas como primitivas, exemplo : execl, execlp, execle, execv e execvp elas permitem a execução de um programa externo ao processo, porém não existe uma criação efetiva de um novo processo e sim a substituição do programa de execução.

A seguir, são apresentadas as sintaxes para cada função do **exec**:

```
int execl( const char *path, const char *arg, ...);

int execlp( const char *file, const char *arg, ...);

int execle( const char *path, const char *arg , ..., char * const envp[]);

int execv( const char *path, char *const argv[]);

int execvp( const char *file, char *const argv[]);
```

Para ser mais claro o parâmetro inicial destas funções é o caminho do arquivo a ser executado, e os parâmetros char arg, são para as funções execl, execlp e execle e podem ser vistos como uma lista de argumentos do tipo arg0, arg1,esses parâmetros descrevem uma lista de um ou mais ponteiros para strings não-nulas.

Segue abaixo um exemplo de um código **exec**:

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

int main(){
   pid_t  pid;
   int ret = 1;
   int status;
   pid = fork();
   if (pid == -1){
      printf("can't fork, error occured\n");
      exit(EXIT_FAILURE);
   }
   else if (pid == 0){
        printf("child process, pid = %u\n",getpid());
        char * argv_list[] = {"ls","-lart","/home",NULL};
        execv("ls",argv_list);
      exit(0);
   }
   else{
      printf("parent process, pid = %u\n",getppid());
        if (waitpid(pid, &status, 0) > 0) {
            if (WIFEXITED(status) && !WEXITSTATUS(status))
              printf("program execution successfull\n");
```

```
else if (WIFEXITED(status) && WEXITSTATUS(status)) {
                    if (WEXITSTATUS(status) == 127) {

                        // execv failed
                        printf("execv failed\n");



      }
                    else
                        printf("program terminated normally,"
                           " but returned a non-zero status\n");

                 }
                 else
                    printf("program didn't terminate normally\n");
              }
              else {
                 // waitpid() failed
                 printf("waitpid() failed\n");
              }
           exit(0);
        }
        return 0;
```

Referencias: Autor desconhecido. 2012. Disponível em: https://daemoniolabs.wordpress.com/tag/como-utilizar-comando-exec/>.

https://www.dca.ufrn.br/~adelardo/cursos/DCA409/node39.html