

# Criação de processos em Unix

A criação de processos em UNIX é feita através das chamadas de sistema `fork` e `execve`. Estes exercícios têm por objetivos mostrar exemplos do uso dessas chamadas e estimular o aluno a compreender sua dinâmica.

## Exercício 1

Dado o programa `fork.c` a seguir:

`fork.c`

```
/*
Criação de processos em UNIX.

Compilar com gcc -Wall fork.c -o fork

Carlos Maziero, DINF/UFPR 2020
*/

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

int main ()
{
    int retval ;

    printf ("Ola, sou o processo %5d\n", getpid() ) ;
    retval = fork () ;
    printf ("[retval: %5d] sou %5d, filho de %5d\n", retval, getpid(),
getppid() ) ;

    if ( retval < 0 ) // erro no fork()
    {
        perror ("Erro") ;
        exit (1) ;
    }
    else
        if ( retval > 0 ) // sou o processo pai
            wait (0) ;
        else // sou o processo filho
            sleep (5) ;

    printf ("Tchau de %5d!\n", getpid() ) ;
    exit (0) ;
}
```

1. Analise seu código e o comente detalhadamente.
2. Desenhe o [diagrama de tempo](#) de sua execução.

## Exercício 2

Dado o programa `fork-execve.c` a seguir:

[fork-execve.c](#)

```
/*
Criação de processos em UNIX, com execução de outro binário

Compilar com gcc -Wall fork-execve.c -o fork-execve

Carlos Maziero, DINF/UFPR 2020
*/

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

int main (int argc, char *argv[], char *envp[])
{
    int retval ;

    printf ("Ola, sou o processo %5d\n", getpid()) ;
    retval = fork () ;
    printf ("[retval: %5d] sou %5d, filho de %5d\n", retval, getpid(),
getppid()) ;

    if ( retval < 0 )          // erro no fork ()
    {
        perror ("Erro: ") ;
        exit (1) ;
    }
    else
        if ( retval > 0 )     // sou o processo pai
            wait (0) ;
        else                  // sou o processo filho
        {
            execve ("/bin/date", argv, envp) ;
            perror ("Erro") ;
        }

    printf ("Tchau de %5d!\n", getpid()) ;
    exit (0) ;
}
```

1. Analise seu código e o comente detalhadamente.
2. Desenhe o [diagrama de tempo](#) de sua execução.
3. O que ocorre se o programa a ser executado por `execve` não existir? Desenhe o diagrama de tempo que retrata essa situação.

## Exercício 3

Dado o programa `fork-print.c` a seguir:

`fork-print.c`

```
/*
Criação de processos em UNIX, com impressão de valores de variável.

Compilar com gcc -Wall fork-print.c -o fork-print

Carlos Maziero, DINF/UFPR 2020
*/

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

int main ()
{
    int retval, x ;

    x = 0 ;
    retval = fork () ;
    printf ("No processo %5d x vale %d\n", getpid(), x) ;

    if ( retval < 0 )
    {
        perror ("Erro") ;
        exit (1) ;
    }
    else
        if ( retval > 0 )
        {
            x = 0 ;
            wait (0) ;
        }
        else
        {
            x++ ;
            sleep (5) ;
        }

    printf ("No processo %5d x vale %d\n", getpid(), x) ;
    exit (0) ;
}
```

1. Analise seu código e o comente detalhadamente.
2. Desenhe o [diagrama de tempo](#) de sua execução, indicando a evolução do valor da variável `x`.

## Observações

- Informações detalhadas sobre as chamadas de sistema utilizadas podem ser encontradas nas páginas de manual do sistema UNIX.
- Um relatório [no formato apropriado](#) deve ser produzido pelo aluno, contendo os resultados obtidos nos exercícios.

From:

<https://wiki.inf.ufpr.br/maziero/> - **Prof. Carlos Maziero**

Permanent link:

[https://wiki.inf.ufpr.br/maziero/doku.php?id=so:criacao\\_de\\_processos](https://wiki.inf.ufpr.br/maziero/doku.php?id=so:criacao_de_processos)

Last update: **2022/11/04 13:00**

