

## Criacao de processos em Unix

### Exercicio 1-

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

int main ()
{
    int retval ; //valor de retorno da criação do processo

    printf ("Ola, sou o processo %5d\n", getpid()) ; //imprime na tela a ID do
    processo, conseguido através da função getpid();

    retval = fork () ;// a função fork() passa o valor de retorno do processo para a
    variável retval;

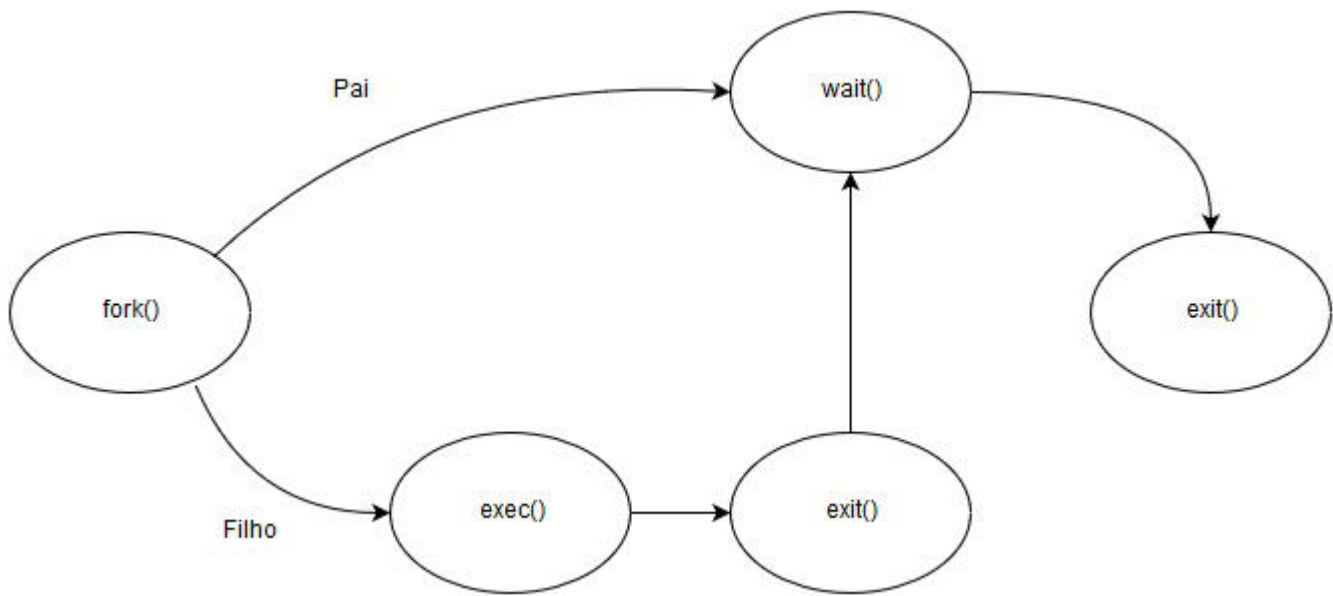
    printf ("[retval: %5d] sou %5d, filho de %5d\n",
            retval, getpid(), getppid()) ; //imprime na tela retval, retornado pela
    função fork(), e a ID do processo filho e a ID do processo pai

    if ( retval < 0 )//checa o valor de retorno
    {
        perror ("Erro") ;//se for negativo, indica erro
        exit (1) ;//termina a execução
    }
    else
    {
        if ( retval > 0 )//se o valor for positivo, significa sucesso na criação
            wait (0) ;
        else //se for exatamente 0, significa sucesso na criação do processo filho
            sleep (5) ;

        printf ("Tchau de %5d!\n", getpid()) ;//encerra com uma mensagem de despedida

        exit (0) ;//termina a execução
    }
}
```

Diagrama do tempo de execução:



Exercicio 2-

```
#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 ; //valor de retorno da criação do processo

    printf ("Ola, sou o processo %5d\n", getpid()) ; //imprime na tela a ID do
    processo, conseguido através da função getpid();

    retval = fork () ;//a função fork() passa o valor de retorno do processo para a
    variável retval;

    printf ("[retval: %5d] sou %5d, filho de %5d\n",
        retval, getpid(), getppid()) ; //imprime na tela retval, retornado pela
    função fork(), e a ID do processo filho e a ID do processo pai

    if ( retval < 0 ) //checa o valor de retorno
    {
        perror ("Erro: ") ; //se for negativo, indica erro
        exit (1) ;//termina a execução
    }
    else
    {
        if ( retval > 0 )//se o valor for positivo, significa sucesso na criação
            wait (0) ;
        else //se for exatamente 0, significa sucesso na criação do processo filho
        {

```

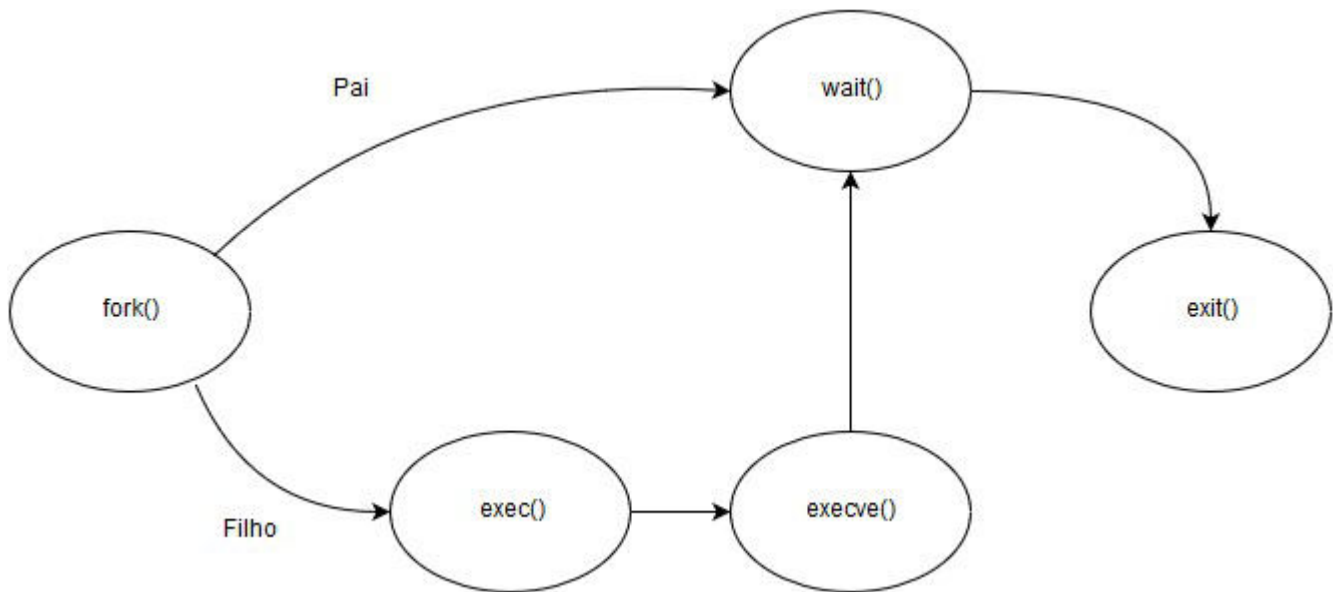
```

        execve ("/bin/date", argv, envp) ;//Executa o processo "/bin/date", que
mostra a hora atual, e a ID do processo filho
        perror ("Erro") ;//Indica erro se o programa executado por execve não
existir
    }

    printf ("Tchau de %5d!\n", getpid()) ;//encerra com uma mensagem de despedida
    exit (0) ;//termina a execução
}

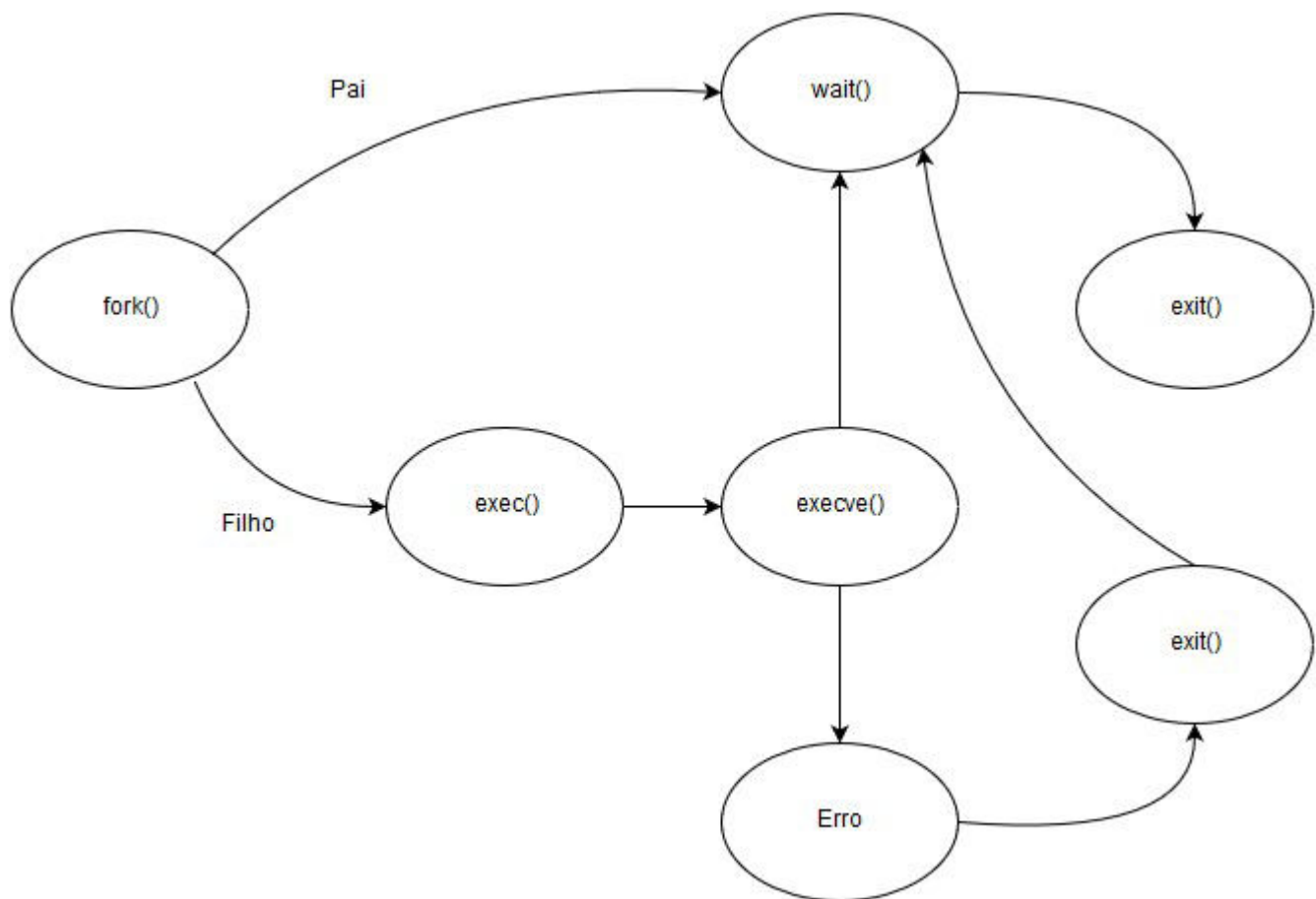
```

Diagrama do tempo de execução:



Se o programa executado por `execve` não existir, mostra a mensagem de erro e retorna o valor -1.

Diagrama do tempo de execução caso ocorra:



Exercicio 3 –

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

int main ()
{
    int retval, x ; //valor de retorno da criação do processo e valor que representa
    o nível hierárquico do processo

    x = 0 ; //inicia x com 0

    retval = fork () ;// a função fork() passa o valor de retorno do processo para a
    variável retval;

    printf ("No processo %5d x vale %d\n", getpid(), x) ;// Mostra a id do processo
    e o valor de X

    if ( retval < 0 ) //checa o valor de retorno
    {
        perror ("Erro") ;//se for negativo, indica erro
        exit (1) ; //termina a execução
    }
}
```

```

}
else
    if ( retval > 0 )//se o valor for positivo, significa sucesso na criação
    {
        x = 0 ;//nivel hierarquico é 0 pois é o processo pai
        wait (0) ;
    }
    else //se for exatamente 0, significa sucesso na criação do processo filho
    {
        x++ ;//Incrementa a variável X para cada nível hierárquico abaixo
        sleep (5) ;
    }

    printf ("No processo %5d x vale %d\n", getpid(), x) ;//Mostra a ID do processo e
o valor de X, representando seu nível hierárquico.

    exit (0) ;//termina a execução
}

```

Diagrama do tempo de execução:

