Lista de Exercícios Práticos

**Criação de Processos**

💻 Laboratório de Sistemas Operacionais

## Introdução

Neste laboratório, você irá praticar a criação e o gerenciamento de processos em um sistema operacional utilizando as chamadas de sistema fork(), wait(), e exec().

Essas chamadas são essenciais para a criação de novos processos, a sincronização entre processos pai e filho, e a substituição do código de um processo em execução por um novo programa.

* **Criação de Processos**: Você irá utilizar a chamada **fork()** para criar novos processos. Quando fork() é chamado, o processo atual (processo pai) cria um novo processo (processo filho), que é uma cópia praticamente idêntica do pai.
* **Sincronização**: A chamada wait() irá garantir que o processo pai aguarde a conclusão do processo filho antes de continuar sua execução.
* **Substituição de Processos**: Por fim, a chamada exec() (e suas variantes) pode ser usada para substituir o processo filho por um novo programa, permitindo que o filho execute comandos diferentes do pai

## Recursos Auxiliares

**Bibliotecas**

|  |
| --- |
| #include <stdio.h>  #include <unistd.h> // fork(), exec()  #include <sys/types.h> // tipos de dados como pid\_t  #include <sys/wait.h> // wait() |

**Chamadas**

* **fork( )**

|  |
| --- |
| pid\_t fork(void); |

* + **No processo pai, retorna o PID do processo filho criado**
  + **No processo filho, retorna 0**
  + **Erro: retorna -1 (e nenhum processo é criado)**
* **wait( )**

|  |
| --- |
| pid\_t wait(int \*status); |

* + **No processo pai, espera até que UM dos seus processos filho terminem.**
  + **O status não é obrigatório**
* **exec( ) e variantes**
  + Recomenda-se usar a variante execlp( ) para o contexto desse laboratório.

|  |
| --- |
| int execlp(const char \*file, const char \*arg, ..., NULL); |

* + **Exemplo de uso**

|  |
| --- |
| execlp("ls", "ls", "-l", "/home", NULL); |

* + Para conhecer mais sobre as variantes do exec( ), consulte os recursos abaixo:
    - <https://www.dca.ufrn.br/~adelardo/cursos/DCA409/node39.html>
    - <https://man7.org/linux/man-pages/man3/exec.3.html>
    - <https://www.ibm.com/docs/en/aix/7.3?topic=e-exec-execl-execle-execlp-execv-execve-execvp-exect-fexecve-subroutine>

**Orientações**

* Lembre de consultar a documentação dos comandos em caso de dúvidas.
* Envie um arquivo .c para cada exercício.

**EXERCÍCIO 1.** Escreva um programa em C que crie um processo filho utilizando fork().

* No processo pai, imprima o PID do processo pai e do filho.
* No processo filho, imprima uma mensagem informando que ele é o filho e o PID dele.
* Adicione um código no processo filho para fazer com que ele execute um loop que imprime uma mensagem 5 vezes, com um intervalo de 1 segundo entre as mensagens.

**EXERCÍCIO 2.** Modifique o programa anterior para que o processo pai use wait() para aguardar que o processo filho termine antes de continuar sua execução.

* Após o término do filho, o pai deve imprimir uma mensagem informando que o filho terminou.

**EXERCÍCIO 3.** Modifique o processo filho para que, em vez de executar um loop, ele execute um novo programa (como ls, ou pwd) usando exec() ou suas outras variações.

* O processo pai deve continuar aguardando o término do filho.