Processos

Eduardo G. R. Miranda

Setembro 2023

O intuito deste documento é de apresentar o conceito de processos filhos e como criar um processo filho na linguagem C.

- O que é um Processo?

  Um processo é um programa em execução, junto com as informações associadas que permitem ao sistema operacional gerenciar e controlar sua execução. Cada processo possui seu próprio espaço de memória, registradores e outros recursos necessários para a execução.

- Processo Pai e Processo Filho:

  Um processo pai é aquele que cria um novo processo, chamado processo filho. O processo pai pode controlar o processo filho, bem como compartilhar informações com ele, se necessário.

- Criação de Processos Filhos:

  A criação de um processo filho geralmente envolve o uso de uma chamada de sistema, como fork() em sistemas Unix-like. A chamada fork() cria uma cópia idêntica do processo pai, incluindo seu estado atual. Essa cópia é o processo filho.

- Independência dos Processos:

  Os processos filhos são independentes uns dos outros e do processo pai. Eles possuem seus próprios espaços de memória, registradores e outras informações de controle. Isso permite que os processos filhos executem tarefas diferentes de forma paralela, melhorando a eficiência e a capacidade de resposta do sistema.

- Comunicação entre Processos:

  Processos filhos podem se comunicar com o processo pai e com outros processos filhos de várias maneiras. Isso pode incluir a troca de informações por meio de variáveis compartilhadas, uso de pipes, filas de mensagens, semáforos, memória compartilhada, entre outros mecanismos de comunicação.

- Terminação de Processos Filhos:

  Quando um processo filho conclui sua tarefa, ele pode terminar sua execução. O processo filho pode retornar um valor de saída para o processo pai usando exit(), indicando seu estado de conclusão. O processo pai pode usar funções como wait() para aguardar a conclusão dos processos filhos e coletar seus estados de saída.

- Concorrência e Paralelismo:

  O uso de processos filhos é fundamental para a execução concorrente e paralela de tarefas em sistemas operacionais. Processos filhos permitem que várias atividades ocorram simultaneamente, melhorando a eficiência e a capacidade de resposta do sistema.

Listing 1: Criação processo filho

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

int fatorial(int n) {
    if (n <= 1) {
        return 1;
    } else {
        return n * fatorial(n - 1);
    }
}

int main() {
    pid_t pid;
    int numero, resultado;

    printf("Digite um numero para calcular o fatorial: ");
    scanf("%d", &numero);

    pid = fork();

    if (pid < 0) {
        fprintf(stderr, "Erro ao criar o processo filho\n");
        return 1;
    } else if (pid == 0) {
        // Codigo do processo filho
        resultado = fatorial(numero);
        exit(resultado); // Termina o processo filho e retorna o
            resultado
    } else {
        // Codigo do processo pai
        wait(&resultado); // Espera o processo filho terminar e
            obtem o resultado
        printf("O fatorial de %d e %d.\n", numero, WEXITSTATUS(
            resultado));
    }
```

```
    return 0;
}
```

Exercicio: Contagem de Números primos:

Neste exercício, você vai criar um programa em C que utiliza processos filhos para contar quantos números primos existem em um intervalo dado.

- O programa principal (processo pai) deve pedir ao usuário para inserir dois números inteiros: inicio e fim, que representam o intervalo no qual você deseja contar os números primos.
- O processo pai deve dividir esse intervalo em subintervalos menores, um para cada processo filho.
- Cada processo filho deve verificar se os números em seu subintervalo são primos.
- Os processos filhos devem retornar a contagem de números primos para o processo pai.
- O processo pai deve somar as contagens recebidas dos processos filhos para obter o total de números primos no intervalo.
- O processo pai deve imprimir o resultado total.