

article [utf8]inputenc graphicx enumitem float amssymb listings color framed tcolorbox
 Revisão da Linguagem C Eduardo G. R. Miranda Agosto 2023
 mygreenrgb0,0.6,0 mygrayrgb0.5,0.5,0.5 mymauvegray0.58,0,0.82 orangergb1,0.64,0 purplergb0.41,0.14,0.62
 customc belowcaptionskip=1breaklines=true, frame=L, xleftmargin=language=C, showstringspaces=false,
 basicstyle=, keywordstyle=green, commentstyle=purple, identifierstyle=blue, stringstyle=orange,
 escapechar=@,style=customc
 document

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

itemize

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.