

Processos & Threads

Othon Luiz Teixeira de Oliveira

Sumário

- 1 Introdução
- 2 Processos
- 3 Exemplo
- 4 Criação e Terminação de Processos
- 5 Threads

Introdução

- Nosso estudo está focado em como os sistemas operacionais são projetados e construídos.

Introdução

- Nosso estudo está focado em como os sistemas operacionais são projetados e construídos.
- O conceito central em qualquer sistema operacional é o processo.

Introdução

- Nosso estudo está focado em como os sistemas operacionais são projetados e construídos.
- O conceito central em qualquer sistema operacional é o processo.
- Um processo é uma abstração de um programa em execução.

Ideia central

Várias coisas ao mesmo tempo

- Enquanto executa um programa do usuário, um computador pode também ler os dados de um disco e mostrar na tela ou enviá-lo a uma impressora

Ideia central

Várias coisas ao mesmo tempo

- Enquanto executa um programa do usuário, um computador pode também ler os dados de um disco e mostrar na tela ou enviá-lo a uma impressora
- Em sistemas multiprogramados a CPU “salta” de programa para programa, executando cada um deles por alguns milissegundos

Ideia central

Várias coisas ao mesmo tempo

- Enquanto executa um programa do usuário, um computador pode também ler os dados de um disco e mostrar na tela ou enviá-lo a uma impressora
- Em sistemas multiprogramados a CPU “salta” de programa para programa, executando cada um deles por alguns milissegundos
- Ou seja; enquanto a cada instante a CPU executa somente um programa, no passar de 1 segundo ela pode ter trabalhado em vários programas.

Ideia central

Várias coisas ao mesmo tempo

- Enquanto executa um programa do usuário, um computador pode também ler os dados de um disco e mostrar na tela ou enviá-lo a uma impressora
- Em sistemas multiprogramados a CPU “salta” de programa para programa, executando cada um deles por alguns milissegundos
- Ou seja; enquanto a cada instante a CPU executa somente um programa, no passar de 1 segundo ela pode ter trabalhado em vários programas.
- $1 \text{ seg.} = 1\,000 \text{ milissegundos} = 1\,000\,000 \text{ microssegundos} = 1\,000\,000\,000 \text{ nanossegundos}$

Modelo de Processo

Nosso Modelo de Processo

- 1 Em Nosso MODELO todos os softwares podem executar em um computador – inclusive o próprio SO

Modelo de Processo

Nosso Modelo de Processo

- 1 Em Nosso MODELO todos os softwares podem executar em um computador – inclusive o próprio SO
- 2 Inclusive o próprio S.O. então um programa (software) pode dar origem a — Processos organizados em vários “processos”

Modelo de Processo

Nosso Modelo de Processo

- ❶ Em Nosso MODELO todos os softwares podem executar em um computador – inclusive o próprio SO
- ❷ Inclusive o próprio S.O. então um programa (software) pode dar origem a — Processos organizados em vários “processos”
- ❸ Um processo é apenas um programa em execução acompanhado dos valores atuais do controlador de processos, dos registradores e das variáveis;

Modelo de Processo

Nosso Modelo de Processo

- 1 Em Nosso MODELO todos os softwares podem executar em um computador – inclusive o próprio SO
- 2 Inclusive o próprio S.O. então um programa (software) pode dar origem a — Processos organizados em vários “processos”
- 3 Um processo é apenas um programa em execução acompanhado dos valores atuais do controlador de processos, dos registradores e das variáveis;
- 4 Conceitualmente cada processo “pensa” ter sua própria CPU;

Modelo de Processo

Nosso Modelo de Processo

- 1 Em Nosso MODELO todos os softwares podem executar em um computador – inclusive o próprio SO
- 2 Inclusive o próprio S.O. então um programa (software) pode dar origem a — Processos organizados em vários “processos”
- 3 Um processo é apenas um programa em execução acompanhado dos valores atuais do controlador de processos, dos registradores e das variáveis;
- 4 Conceitualmente cada processo “pensa” ter sua própria CPU;
- 5 Na realidade a CPU troca, a todo momento, de um processo para outro.

Alternância entre Processos

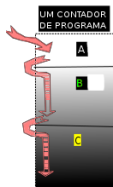


Figura: Multi-
programação de
4 processos

Alternância entre Processos

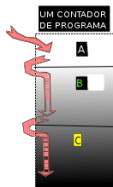


Figura: Multi-programação de 4 processos

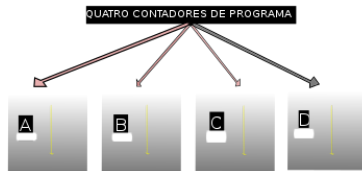


Figura: Modelo conceitual de 4 processos sequenciais independentes

Contador de Programas

Contador de Programas

- Na primeira figura vimos um computador multiprogramado com 4 processos na memória.

Contador de Programas

Contador de Programas

- Na primeira figura vimos um computador multiprogramado com 4 processos na memoria.
- Na segunda figura estão 4 processos, cada um com seu próprio fluxo de controle, isto é; seu próprio contador de programas (lógico)

Contador de Programas

Contador de Programas

- Na primeira figura vimos um computador multiprogramado com 4 processos na memoria.
- Na segunda figura estão 4 processos, cada um com seu próprio fluxo de controle, isto é; seu próprio contador de programas (lógico)
- Contudo só há um contador de programa real

Contador de Programas

Contador de Programas

- Na primeira figura vimos um computador multiprogramado com 4 processos na memória.
- Na segunda figura estão 4 processos, cada um com seu próprio fluxo de controle, isto é; seu próprio contador de programas (lógico)
- Contudo só há um contador de programa real
- Quando acaba o tempo da CPU alocado para um processo, o contador de programas físico é salvo no contador de programas lógico, do processo em memória.

Exemplo

O cientista confeitiro

- Imagine um cientista em computação com seus dotes culinários vai fazer um bolo de aniversário para sua filha.

Exemplo

O cientista confeitiro

- Imagine um cientista em computação com seus dotes culinários vai fazer um bolo de aniversário para sua filha.
- Ele tem a receita e uma cozinha bem equipada, com todos os ingredientes: farinha, ovos, açúcar, essência de coco, entre outros.

Exemplo

O cientista confeitiro

- Imagine um cientista em computação com seus dotes culinários vai fazer um bolo de aniversário para sua filha.
- Ele tem a receita e uma cozinha bem equipada, com todos os ingredientes: farinha, ovos, açúcar, essência de coco, entre outros.
- Nessa analogia a receita é o programa(o algoritmo), os ingredientes são as entradas (os inputs) e o processador é o cientista. Pergunta: E o processo ???

Exemplo

O cientista confeitiro

- Imagine um cientista em computação com seus dotes culinários vai fazer um bolo de aniversário para sua filha.
- Ele tem a receita e uma cozinha bem equipada, com todos os ingredientes: farinha, ovos, açúcar, essência de coco, entre outros.
- Nessa analogia a receita é o programa(o algoritmo), os ingredientes são as entradas (os inputs) e o processador é o cientista. Pergunta: E o processo ???
- Agora imagine que o outro filho do cientista chegue chorando; cortou o dedo. O cientista registra onde estava na receita (estado atual do processo é salvo)

Exemplo - cont...

O cientista confeitreiro

Exemplo - cont...

O cientista confeitreiro

- Pega um livro de primeiros socorros e segue as instruções contidas nele. Nesse ponto o processador está sendo alternado de um processo (assar o bolo) para outro processo de maior prioridade (socorrer o outro filho).

Exemplo - cont...

O cientista confeitiro

- Pega um livro de primeiros socorros e segue as instruções contidas nele. Nesse ponto o processador está sendo alternado de um processo (assar o bolo) para outro processo de maior prioridade (socorrer o outro filho).
- A ideia principal é que um processo constitui uma atividade. Ele possui um programa, entrada, saída e um estado.

Exemplo - cont...

O cientista confeitreiro

- Pega um livro de primeiros socorros e segue as instruções contidas nele. Nesse ponto o processador está sendo alternado de um processo (assar o bolo) para outro processo de maior prioridade (socorrer o outro filho).
- A ideia principal é que um processo constitui uma atividade. Ele possui um programa, entrada, saída e um estado.
- Cada processador pode ser compartilhado entre vários processos, com algum algoritmo de escalonamento, usado para determinar quando parar o trabalho sobre um processo e servir um outro.

Criação de processos

Existência de processos

- O sistema operacional precisa de alguma maneira assegurar todos os processos necessários.

Criação de processos

Existência de processos

- O sistema operacional precisa de alguma maneira assegurar todos os processos necessários.
- Um forno de micro-ondas é um sistema simples ?

Criação de processos

Existência de processos

- O sistema operacional precisa de alguma maneira assegurar todos os processos necessários.
- Um forno de micro-ondas é um sistema simples ?
- Em sistemas simples (embarcados ??) que executam apenas uma única aplicação, é possível que todos os processos sejam iniciados quando o sistema é ligado

Sistemas de propósito geral

Há 4 eventos principais que fazem processos serem criados

1. Início do sistema.
2. Execução de uma chamada ao sistema de criação de processo por um processo em execução
3. Uma requisição do usuário para criar um novo processo.
4. Início de um job em lote.

Sistemas de propósito geral

Depois de criado um processo começa a executar e faz seu trabalho. Contudo nem mesmo os processos são para sempre. Mais cedo ou mais tarde em razão das seguintes condições:

Condições de termino

1. Saída normal (voluntária).
2. Saída por erro (voluntária).
3. Erro fatal (involuntário).
4. Cancelamento por um outro processo (involuntário).

Sistemas de propósito geral

Em alguns sistemas, quando um processo cria outro processo, o processo pai e o processo filho, de certa forma permanecem associados. O próprio processo filho pode criar outros processos, gerando uma hierarquia de processos

Unix

No Unix um processo pai e todos os processos filhos formam um grupo de processos. No Windows não há nenhum conceito de hierarquia de processos. Algo parecido ocorre quando um processo cria um processo “pai” (chamado handle) pode usar para controlar o filho, contudo ele é livre para passar esse identificador para outro processo.

Os processos no Unix não podem deserdar seus filhos.

Estados de um processo

Embora cada processo seja uma entidade independente, com seu contador de programas, muitas vezes um processo precisam interagir com outros processos.

Um processo pode gerar uma saída que outro processo usa como entrada

Comando

```
cat capitulo1 capitulo2 capitulo3 — grep tree
```

O processo cat concatena, para a saída padrão três processos. Dependendo das velocidades que dois processos (CPU, complexidade do processo) pode acontecer que o grep esteja pronto antes do car. O processo bloqueia até que o cat tenha terminado

Estados de um processo – simplificado

Como vimos na aula anterior um processo pode estar em vários estados

Estados principais de um processo

1. Em execução (realmente usando a CPU naquele instante)
2. Pronto (executável; temporariamente parado para dar lugar a outro processo).
3. Bloqueado (incapaz de executar enquanto um evento externo não ocorrer).

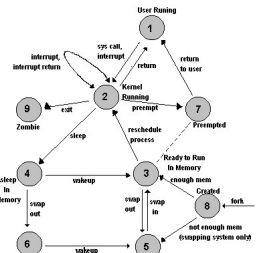
Estado simplificado dos processos

Os dois primeiros estados são similares (Em execução e Pronto). Em ambos os casos o processo vai executar, só que no segundo não há CPU disponível para ele. O terceiro estado é diferente dos dois primeiros, pois o processo não pode executar, mesmo que a CPU não tenha nada pra fazer.



Estado simplificado dos processos

Os dois primeiros estados são similares (Em execução e Pronto). Em ambos os casos o processo vai executar, só que no segundo não há CPU disponível para ele. O terceiro estado é diferente dos dois primeiros, pois o processo não pode executar, mesmo que a CPU não tenha nada pra fazer.



Espaço de endereçamento

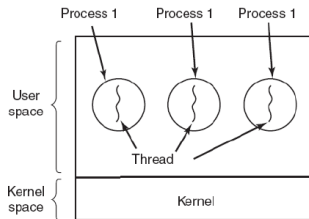
Em sistemas tradicionais, cada processo tem um espaço de endereçamento e um único Thread (fluxo) de controle.

Definição

Isso é quase uma definição de processo, exceto pelo espaço de endereçamento

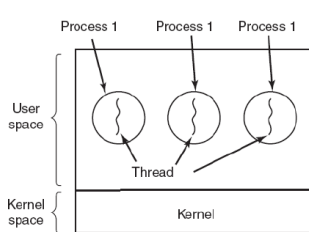
Contudo é frequente querer ter múltiplos threads em um único espaço de endereçamento executando em quase-paralelamente, como se fossem processos separados.

Um processo com múltiplas threads

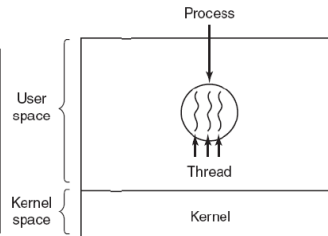


(a)

Um processo com múltiplas threads



(a)



(b)

Processos Preemptivos e Não-Preemptivos

Estados principais de um processo

Pesquisar o que são esses tipos de processos (Preemptivos e Não-Preemptivos), descrevendo cada algoritmo que existe.

Estados principais de um processo

Assistam o vídeo da Aula 4 para entender o que pesquisar