

Ciclo de Vida de um Processo

Marcelo Johann

Nas aulas anteriores...

- Conceito de Processo
- Mecanismo de Interrupção
- Programação em C/UNIX

- Continuando
 - Interrupções
 - TRAP
 - Chaveamento de Contexto
 - Chamadas de Sistema

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 2

Plano da aula

- Definição do processo (relembrando)
- Interrupções (relembrando)
- Chamadas de Sistema
- Criação e Comunicação entre processos
 - fork
 - pipes
 - sinalização
- Imagem do processo
- Ciclo de vida do processo
 - Estados e transições

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 3

O conceito de processo

- Um **programa** é:
 - Uma seqüência finita de instruções;
 - Uma entidade **passiva** (que não se altera com o passar do tempo).
 - Armazenado em disco.
- Um **processo** é:
 - Uma abstração que representa um programa **em execução**;
 - Uma entidade **dinâmica**: seu estado se altera conforme for executando.
 - Armazenado na memória.
- Pode-se encontrar mais de um processo instanciando um programa único.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 4

O processo do ponto de vista do S.O.

- Imagem de um programa
 - Segmento de código
 - Espaço de endereçamento
- **Conjunto de recursos** de HW alocados pelo Sis. Op.:
 - Registradores (PC, *Stack Pointer*...);
 - Memória;
 - Espaço no disco (arquivos de E/S).
 - = **Contexto** do processo.
- **Unidade de escalonamento**
 - Estado;
 - Algoritmos de escalonamento para otimizar o uso do HW.
 - Alocar a CPU a um processo implica em uma **troca de contexto**

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 5

Tabela de Processos

- Para manter as informações relativas aos processos, o núcleo deve manter:
 - Uma estrutura de dados relativa a um dado processo
 - struct proc
 - Process Control Block (PCB)
 - Uma estrutura de dados que gerencia o conjunto de processos
 - Tabela de processos.
- As chamadas de sistema que manipulam os processos irão interagir com essas estruturas de dados.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 6

Relacionamento entre processos

- Caso mais simples: os processos são independentes.
 - Sem relacionamento
- Grupo de processos
 - Compartilhamento de recursos
 - Baseados em **hierarquia** de processos:
 - Um processo pai cria processos filhos;
 - Os filhos podem executar o mesmo código, ou trocá-lo;
 - Obtem-se uma árvore de processos.
- Implica na definição da semântica de termino de um processo:
 - Só o processo morre;
 - Toda sua descendência morre.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 7

Suporte de Hardware: interrupções

- Erros e eventos são **detectados por hardware**
 - Exemplos de evento: inserir um pendrive na porta USB, escrever um bloco em disco, receber um pacote pela rede, escrever numa área proibida...
 - O HW emite uma interrupção.
- São **tratados pelo Sis. Op.**
 - Identifica a interrupção (número);
 - Verifica sua prioridade;
 - Acha no vetor de interrupções qual procedimento é apropriado (*handler*).

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 8

Suporte de Hardware: modos de execução

- O HW provê no mínimo dois modos de execução diferentes:
 - Modo **privilegiado** (protegido, sistema...) – todo o conjunto de operações é disponível. É o modo de execução do Sis. Op.
 - Modo **usuário**: uso limitado. Os processos usuários operam neste modo.
- **Chaveamento** de modos:
 - É o fato de passar de um modo para o outro.
 - Usuário -> protegido: por interrupção.
 - Protegido -> usuário: por instrução clássica (yield, return, ...).

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 9

Exemplo importante de uso dos 2 modos

- Para proteger os periféricos, as instruções de E/S são privilegiadas.
- Logo, um processo usuário não pode acessá-los
 - E.g.: escrita em disco, leitura de um CD...
- O usuário deve passar pelo Sis. Op. através de uma chamada de sistema, que gera uma interrupção (trap).



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 10

Chamada de sistema com interrupção

- A chamada de sistema oferece o serviço ao processo usuário, em modo "seguro".
- Ela gera uma interrupção
 - Identificação, prioridade, *handler*...
- Ela implica em uma **troca de contexto**
 - O processo chamador deve deixar o lugar para o código do núcleo!
 - A troca de modo implica em uma troca de contexto.
- Conforme for a prioridade e o tipo de escalonador, a troca de contexto pode ser imediata ou atrasada.
- O que acontece se houver uma interrupção durante o tratamento de uma interrupção?
 - Comparam-se as prioridades
 - Possibilidade de desabilitar as interrupções em casos críticos.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 11

Processos e Disco

- Os processos devem interagir com o disco para armazenar e recuperar dados não voláteis.
- O disco físico é abstraído pelo Sistemas de Arquivos, de acordo com uma hierarquia:
 - Diretórios
 - Arquivos.
- Os diretórios estão freqüentemente organizados de acordo com uma hierarquia em árvore
 - Raiz ('/')
 - Diretório de trabalho de um processo ('.')
 - Caminho relativo / absoluto
- Deve ter chamadas de sistema para acessar o Sistema de Arquivos!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 12

Chamadas de sistema

- A chamada de sistema oferece o serviço ao processo usuário, em modo “seguro”.
- Ela gera uma interrupção
 - Identificação, prioridade, *handler*...
- Ela implica em uma **troca de contexto**
 - O processo chamador deve deixar o lugar para o código do núcleo!
 - A troca de modo implica em uma troca de contexto.
- Conforme for a prioridade e o tipo de escalonador, a troca de contexto pode ser imediata ou atrasada.
- O que acontece se houver uma interrupção durante o tratamento de uma interrupção?
 - Comparam-se as prioridades
 - Pode desabilitar as interrupções em casos críticos.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 13

Exemplos de chamadas de sistema

- Minix 2 provê 53 chamadas de sistema no total
- Gerenciamento de processos:
 - `fork()`, `waitpid()`, `exit()`, `execve(...)`, `getpid()`...
- Sinais
 - `sigaction()`, `sigreturn()`, `sigprocmask()`, `kill()`...
- Gerenciamento de arquivos
 - `open()`, `close()`, `mknod()`, `read()`, `write()`, `pipe()`,...
 - `mkdir()`, `mount()`,...
- Direitos de acesso
- Gerenciamento de tempo
 - `Time()`

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 14

Chamadas de sistema (Linux) para gerenciar processos

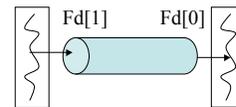
- Criação de processo: **fork()**
 - Cria um novo processo
 - Igual ao pai (clone!)
 - No processo pai, `fork()` retorna o pid do filho;
 - No processo filho, `fork()` retorna 0.
- Mudar o segmento de código: **exec()**
 - Executa o binário apontado em argumento.
 - Em geral, chamado logo após o `fork()` (“fork-exec”)
- Recuperar o identificador: **getpid()**
 - Retorna um int, que identifica o processo.
- Terminar o processo: **kill()**
 - Manda um sinal (e.g. TERM) para o processo cujo pid é dado em argumento.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 15

Comunicação entre processos (IPC)

- Uma forma de fazer com que processos se comuniquem: **pipe()**
- Criam-se dois descritores de arquivos
 - `int fd[2]; pipe(fd);`
- Um serve para a escrita, o outro para ler...
- Usando 2 processos, um pode escrever no *pipe*, e o outro ler do *pipe*...



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 16

Sinalização de processos

- Uma das formas de interagir entre processos é através de sinais.
 - A recepção é assíncrona
 - Ao receber um sinal, o processo para sua atividade,
 - Ele executa um tratamento de sinal adaptado (*signal handler*),
 - Ao se encerrar o tratamento, o processo pode voltar ao estado onde estava antes.
- Um sinal é uma versão em nível de software das interrupções de hardware.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

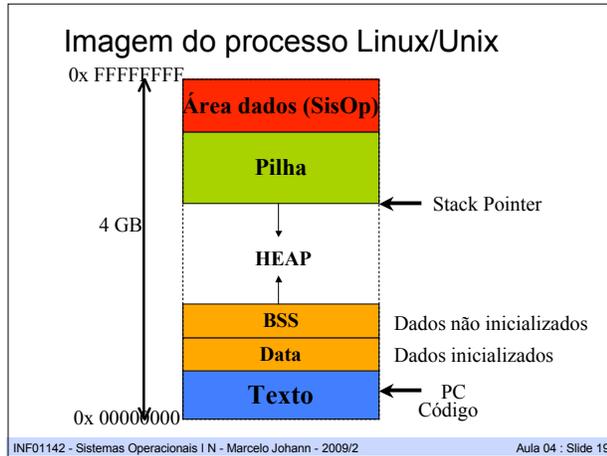
Aula 04 : Slide 17

Representação da imagem do processo

- O processo é representado por uma “imagem”:
 - Seu segmento de código (o que ele vai fazer)
 - Seu espaço de endereçamento (onde, na memória, ele vai fazer alguma coisa)
 - Seu contexto (o que que ele precisa para fazer alguma coisa)
 - Cf. PCB
- Parte da imagem está sob a responsabilidade do usuário; parte dela só é gerenciada em modo protegido (pelo Sis. Op.)

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 04 : Slide 18

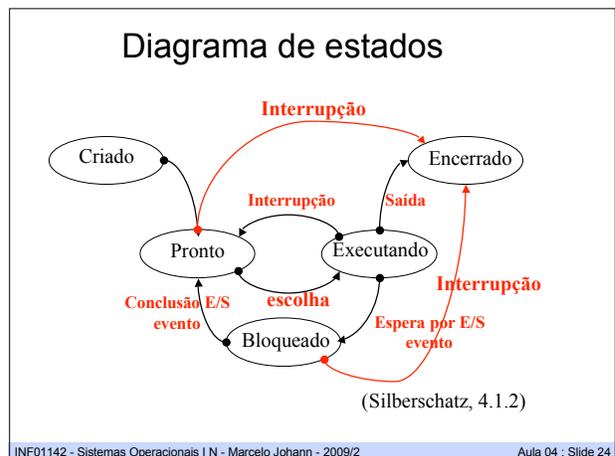


- ### Ciclos e tipos de processo
- Enquanto executam, os processos apresentam 2 tipos de comportamento:
 - Ou usam a CPU
 - Cálculo, operações sobre a memória/registadores
 - Ou fazem Entrada/saída – liberam a CPU
 - Escrita na tela, entrada teclado, som, vídeo, rede, disco...
 - Fala-se de processos **CPU-bound** vs. **I/O-bound**
 - Caso perfeito: **mesclar** os dois tipos de CPU através de escalonamento.
 - Na real: difícil definir qual processo é CPU/IO – bound!
- INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 04 : Slide 20

- ### Ciclo de vida – estados do processo
- Processos **nascem**
 - No momento de sua criação (via chamada de sistema – Fork, spawn...)
 - Processos **vivem**
 - Executam na CPU, liberam a CPU (E/S)...
 - Executam:
 - Programas dos usuários
 - Programas do sistema (*daemons*)
 - Processos **morrem**
 - Ou porque terminaram sua execução
 - Ou porque um outro processo os matou:
 - Erro, acesso não-autorizado, falha
- INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 04 : Slide 21

- ### Ciclo de vida – troca de estado
- Os processos evoluem
 - Criado -> "em vida" -> morto
 - Durante a vida, passam de "uso de CPU" e a "uso de E/S" (e reciprocamente)
 - Via **chamada de sistema**, **interrupção**, ou por causa de um **evento**.
 - Ao ser criado, um processo estará **pronto** para usar a CPU.
 - O que acontece se a CPU não está disponível?
 - O que acontece se vários processos estão sendo criados no mesmo tempo?
 - Precisa-se manter uma lista de processos prontos!
- INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 04 : Slide 22

- ### Ciclo de vida – troca de estado (cont.)
- Ao executar, o processo pode querer fazer E/S:
 - O que acontece se o recurso de E/S está sendo ocupado?
 - É preciso de uma fila de processos **bloqueados**
 - Após ter executado, o processo passa a ser **encerrado**.
 - Em resumo:
 - 5 estados (criado, pronto, executando, bloqueado, encerrado)
 - Transições possíveis.
- INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 04 : Slide 23



O que causa uma transição?

- Pronto -> executando
 - Algoritmo de escalonamento
- Executando -> pronto
 - Interrupção de tempo
 - Interrupção devida ao escalonador
 - Decisão espontânea (yield)
- Executando -> bloqueado
 - E/S – sincronização
- Bloqueado -> pronto
 - Interrupção
- Executando -> encerrado
 - Interrupção (Ctrl-C)
 - Término normal
- Bloqueado, pronto -> encerrado
 - Interrupção (Ctrl-C, kill)

Mais um estado: suspenso

- Dois problemas principais para gerar os recursos:
 - A CPU é muito mais rápida do que a memória;
 - A memória é de tamanho finito.
- Precisa-se, então, **poupar a memória**.
- Por isso:
 - Processos bloqueados que estão na memória podem ser transferidos para o disco (**swap**) até sua E/S ser acabada.
 - Processos prontos podem também ser descarregados para o disco.
- Chega-se a mais dois estados:
 - Bloqueado, suspenso.
 - Pronto, suspenso.

Próxima aula

Implementação de Processos