

Lprm
Laboratório de Pesquisa em Redes e Multimídia



Processos
(Aula 5)

Estruturas de Controle

UFES Universidade Federal do Espírito Santo
Departamento de Informática

Lprm Laboratório de Pesquisa em Redes e Multimídia

Aula passada

- Interrupções de hw e sw (mudança de fluxo, podendo gerar transição da execução entre processos)
- Gerência de Processos
 - O que são processos
 - Abstração de processos no SO
 - Criação de um processo no Unix (fork)

Profª. Patrícia D. Costa LPRM/DI/UFES 2 Sistemas Operacionais 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Aula passada (2)

- Estados do processo
 - Novo (*new*), pronto (*ready*), bloqueado ou esperando (*blocked*), executando (*running*), terminado (*exit*)
- Transição entre os estados
- Filas do sistema
- Máquina de estados do UNIX

Profª. Patrícia D. Costa LPRM/DI/UFES 3 Sistemas Operacionais 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Processos

- O S.O. controla eventos no sistema de computação:
 - S.O. escalona e despacha processos para execução, aloca os recursos para processos, e responde a pedidos de processos de usuários a serviços básicos
- S.O. é a entidade que gerencia o uso de recursos do sistema pelos processos.

Profª. Patrícia D. Costa LPRM/DI/UFES 4 Sistemas Operacionais 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Processos e Recursos (1)

Figure 3.9 Processes and Resources (resource allocation at one snapshot in time)

Profª. Patrícia D. Costa LPRM/DI/UFES 5 Sistemas Operacionais 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Processos e Recursos (2)

- O S.O. gerencia recursos computacionais em benefício dos diversos processos que executam no sistema.
- A questão fundamental é:
 - Que informações o sistema operacional precisa manter para controlar os processos e gerenciar os recursos em benefícios deles?

Profª. Patrícia D. Costa LPRM/DI/UFES 6 Sistemas Operacionais 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Tabelas de Controle do S.O. (1)

- O S.O. precisa manter informações sobre o estado corrente de cada processo e de cada recurso.
- Estruturas de controle são construídas para cada entidade que o sistema operacional gerencia:
 - Tabelas de Memória
 - Tabelas de E/S
 - Tabelas de Arquivos
 - Tabelas de Processos

Profª. Patrícia D. Costa LPRM/DI/UFES 7 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Tabelas de Controle do S.O. (2)

Figure 3.10 General Structure of Operating System Control Tables

Profª. Patrícia D. Costa LPRM/DI/UFES 8 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Estruturas de Controle de Processos

- Multiprogramação pressupõe a existência de vários processos disputando a UCP.
- Necessidade de intermediar essa disputa.
 - Técnicas de gerência do processador.
 - Algoritmos de escalonamento.
- Necessidade de representar um processo.
 - *Imagem* do processo
 - Localização: parte da imagem do processo na memória primária e parte na secundária.

Profª. Patrícia D. Costa LPRM/DI/UFES 9 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Imagem do Processo (1)

- Nome dado à coleção formada por:
 - Código do programa a ser executado.
 - Pilha do sistema para controle de chamadas de procedimentos e de SVCs.
 - Área de dados para armazenamento de variáveis locais e globais.
 - Coleção de atributos do processo (mantidos no *Bloco de Controle de Processos*).

Profª. Patrícia D. Costa LPRM/DI/UFES 10 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Imagem do Processo (2)

Profª. Patrícia D. Costa LPRM/DI/UFES 11 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Bloco de Controle de Processo (atributos)

- Estrutura de dados (registro) usada para representar um processo dentro do sistema operacional.
- Mantém todas as informações que o S.O. precisa para poder controlar a execução do processo (coleção de atributos do processo).
- Número fixo ou variável de blocos descritores de processos (alocação estática x alocação dinâmica de memória).
- *Process Control Block*.

Profª. Patrícia D. Costa LPRM/DI/UFES 12 Sistemas Operacionais: 2008/1

Informações Típicas do BCP

- Identificação do processo.
- Prioridade do processo.
- Localização na memória principal.
- Identificação dos arquivos abertos.
- Estado do processo.
- Contexto de execução (conteúdo dos registradores).
- Ponteiros para encadeamento nas filas.
- Informações de *accounting* (ex: tempo de CPU).

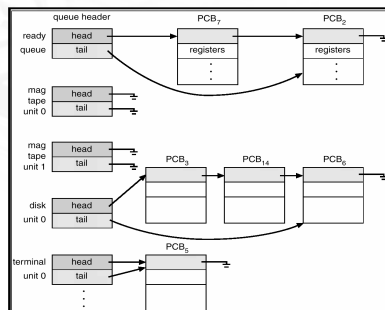
Estrutura de Dados Típica

```

struct desc_proc {
    char        estado_atual;        /* Estado atual do processo */
    int         prioridade;          /* Prioridade do processo */
    unsigned    inicio_memoria;     /* Endereço inicial da memória */
    unsigned    tamanho_memoria;    /* Bytes de memória ocupados */
    struct      arquivo_arq_abertos[20] /* Arquivos abertos */
    unsigned    tempo_de_CPU;       /* Tempo já gasto de CPU */
    unsigned    proc_pc;            /* Valor salvo do registrador PC */
    unsigned    proc_sp;            /* Valor salvo do registrador SP */
    unsigned    proc_acc;           /* Valor salvo do registrador ACC */
    unsigned    proc_rx;            /* Valor salvo do registrador RX */
    struct      desc_proc *proximo  /* Aponta para o próximo */
}
struct desc_proc tab_desc[MAX_PROCESS];

```

BCPs e as Filas do Sistema



Inicialização das Filas

```

struct desc_proc *desc_livres;      /* Lista de descritores livres */
struct desc_proc *espera_CPU;     /* Lista de processos prontos */
struct desc_proc *usando_CPU;     /* Aponta p/ processo em execução */
/*
struct desc_proc *bloqueados;     /* Lista de processos bloqueados */

/* Inicialização das estruturas de controle */

for (i=0; i<MAX_DESC_PROC - 1; ++i)
    tab_desc[i].proximo = &tab_desc[i+1];

tab_desc[i].proximo = NULL;
desc_livres = &tab_desc[0];
espera_CPU = NULL;
usando_CPU = NULL;
bloqueados = NULL;

```

Tipos de Informações do BCP

- As informações mantidas no BCP podem ser divididas em três categorias:
 - Identificação do processo;
 - Informações de estado do processador;
 - Informações de controle do processo.

Identificação do Processo

- Os seguintes identificadores podem estar armazenados no BCP :
 - Identificador do processo;
 - Identificador do processo que o criou (processo pai);
 - Identificador do usuário.

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Informações de Estado da UCP (1)

- Registradores visíveis ao usuário
 - Aqueles que podem ser referenciados por meio da linguagem de máquina que o processador executa.
- Registradores de controle e estado
 - *Program Counter* (PC);
 - *Flags*: sign, zero, carry, equal, overflow;
 - *Status*: modo supervisor x usuário, interrupção habilitada x desabilitada.
 - *Stack Pointers*.

Profª. Patrícia D. Costa LPRM/DI/UFES 19 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Informações de Estado da UCP (2)

Figure 3.11 Pentium II EFLAGS Register

ID = Identification flag	DF = Direction flag
VIP = Virtual interrupt pending	IF = Interrupt enable flag
VIF = Virtual interrupt flag	TF = Trap flag
AC = Alignment check	SF = Sign flag
VM = Virtual 8086 mode	ZF = Zero flag
RF = Resume flag	AF = Auxiliary carry flag
NT = Nested task flag	PF = Parity flag
IOPL = I/O privilege level	CF = Carry flag
OF = Overflow flag	

Profª. Patrícia D. Costa LPRM/DI/UFES 20 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Informações de Controle do Processo (1)

- Informações de Escalonamento e Estado:
 - Estado do processo (*ready, running, suspended, etc.*)
 - Prioridade (*default, corrente, máxima*)
 - Tempo de espera na fila
 - Tempo de execução na última fatia de tempo
 - Evento que o processo está aguardando
- Estruturação de dados:
 - Um processo pode estar encadeado a outros em uma fila, lista ou outra estrutura de dados.

Profª. Patrícia D. Costa LPRM/DI/UFES 21 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Informações de Controle do Processo (2)

- Comunicação entre processos:
 - Flags, sinais e mensagens podem estar associados com a comunicação entre dois processos independentes.
 - Algumas ou todas essas informações podem estar mantidas no BCP.
- Privilégios em termos de memória que pode ser acessada, instruções que podem ser executadas, ou mesmo serviços e utilitários do sistema.

Profª. Patrícia D. Costa LPRM/DI/UFES 22 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Informações de Controle do Processo (3)

- Gerência de Memória:
 - Ponteiros para tabelas de páginas ou de segmentos que descrevem a memória virtual assinalada ao processo.
- *Ownership* e utilização de recursos:
 - Arquivos abertos;
 - Histórico de uso da UCP ou de outro recurso (para usos do escalonador);

Profª. Patrícia D. Costa LPRM/DI/UFES 23 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

O Escalonador ("Scheduler")

- Módulo do S.O. responsável pelo controle do recurso "processador".
- Divide o tempo da UCP entre os processos do sistema.
- Três tipos básicos:
 - Escalonador de curto prazo ("*short-term scheduler*");
 - Escalonador de longo prazo ("*long-term scheduler*");
 - Escalonador de médio prazo ("*medium-term scheduler*").

Profª. Patrícia D. Costa LPRM/DI/UFES 24 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Escalonador de Curto Prazo (1)

- Escalonador da UCP
 - Dispatcher, CPU Scheduler
- Seleciona qual processo deve ser executado a seguir (*ready*→*running*).
- É invocado muito frequentemente (ordem de milissegundos). Deve, portanto, ser rápido.
- É invocado sempre que um "evento bloqueador" ou "evento preemptivo" ocorrer
 - Interrupção do clock, Interrupção de I/O, Chamada de sistema, Sinais (e.x., semáforos)

Profª. Patrícia D. Costa LPRM/DI/UFES 25 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Escalonador de Curto Prazo (2)

The diagram illustrates the state transitions in a short-term scheduler. It shows a 'ready queue' that feeds into the 'CPU'. From the 'CPU', processes can move to 'I/O request', 'time slice expired', 'fork a child', or 'wait for an interrupt'. 'I/O request' leads to an 'I/O queue' and then back to 'I/O'. 'time slice expired' leads back to the 'ready queue'. 'fork a child' leads to 'child executes'. 'wait for an interrupt' leads to 'interrupt occurs', which then feeds back into the 'ready queue'.

Profª. Patrícia D. Costa LPRM/DI/UFES 26 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Escalonador de Longo Prazo

- Escalonador de *Jobs* ("Job Scheduler").
- Seleciona quais processos devem ser levados para a fila de prontos (*new*→*ready*).
- Baixa frequência de invocação (ordem de segundos ou minutos).
- Permite o controle da carga no sistema, (controla o grau de multiprogramação).
- Decisão para aceitar processos:
 - equilíbrio entre I/O bound e CPU bound processes;
 - equilíbrio de uso de serviços de I/O.

Profª. Patrícia D. Costa LPRM/DI/UFES 27 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Escalonador de Médio Prazo (1)

- Utiliza a técnica de *swapping*.
 - Swap out*: a execução do processo é suspensa e o seu código e dados são temporariamente copiados para o disco.
 - Swap in*: o processo é copiado de volta do disco para a memória e sua execução é retomada do ponto onde parou.
- Está intimamente ligado à gerência de memória.

Profª. Patrícia D. Costa LPRM/DI/UFES 28 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Escalonador de Médio Prazo (2)

The diagram shows the interaction between swapped-out states and the ready queue. 'partially executed swapped-out processes' can be 'swap in' to the 'ready queue' or 'swap out' back to the swapped-out state. The 'ready queue' feeds into the 'CPU'. From the 'CPU', processes can go to 'end', 'I/O waiting queues', or back to the 'ready queue'. 'I/O waiting queues' leads to 'I/O', which then feeds back into the 'ready queue'.

Profª. Patrícia D. Costa LPRM/DI/UFES 29 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia

Troca de Contexto

- Contexto* de um processo são todas as informações necessárias para que o S.O. possa restaurar a execução do processo a partir do ponto interrompido.
- A troca de contexto ocorre sempre que um novo processo é selecionado para execução (isso é, quando a UCP é chaveada para um outro processo).
- O tempo de troca de contexto é puro *overhead* e é dependente de suporte de hardware (ex: salvamento automático do PC).

Profª. Patrícia D. Costa LPRM/DI/UFES 30 Sistemas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Quando Chavear um Processo

- Interrupção do relógio
 - Fatia de tempo de posse da UCP expirou;
- Interrupção de E/S
- Falta de memória
 - Endereço de memória está na memória virtual (disco); logo deve ser trazido para a memória principal.
- Trap
 - Ocorrência de erro.
 - Pode fazer com que o processo seja movido para o estado *Exit*.

Profª. Patrícia D. Costa LPRM/DI/UFES 31 Siste.mas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Ações na Troca de Contexto (1)

- Salvar o contexto do processador, incluindo o PC e outros registradores.
- Alterar o BCP do processo que está no estado "em-execução" (*running*).
- Mover o BCP para a fila apropriada.
- Selecionar outro processo para execução.
- Alterar o BCP do processo selecionado.
- Alterar as tabelas de gestão de memória.
- Restaurar o contexto do processo selecionado.

Profª. Patrícia D. Costa LPRM/DI/UFES 32 Siste.mas Operacionais: 2008/1

Lprm Laboratório de Pesquisa em Redes e Multimídia UFES

Ações na Troca de Contexto (2)

The diagram illustrates the context switching process between two processes, P_0 and P_1 , through the operating system. It shows the state of each process (executing or idle) and the actions performed by the OS: saving state into PCBs and reloading state from PCBs.

Process P_0 starts in an "executing" state. An "interrupt or system call" occurs, leading to the OS performing "save state into PCB₀". Process P_0 then becomes "idle". The OS then performs "reload state from PCB₁", and process P_1 becomes "executing". This cycle repeats: P_1 is interrupted, its state is saved to PCB₁, it becomes idle, and then P_0 's state is reloaded from PCB₀, making P_0 "executing" again.

Profª. Patrícia D. Costa LPRM/DI/UFES 33 Siste.mas Operacionais: 2008/1