Sistemas Operacionais

Processos

O conceito de processos

- No capítulo 1, fizemos as seguintes afirmativas quanto aos processos:
  - Mais geral que programa
  - Consiste em um código executável e seus dados associados, além de um contexto de execução
  - Tudo em sistemas atuais está em torno de processos

Enfim, ...

- Um processo é uma abstração de um programa em execução
- Exemplo:
  - fazer um bolo ...
  - ... primeiros socorros!

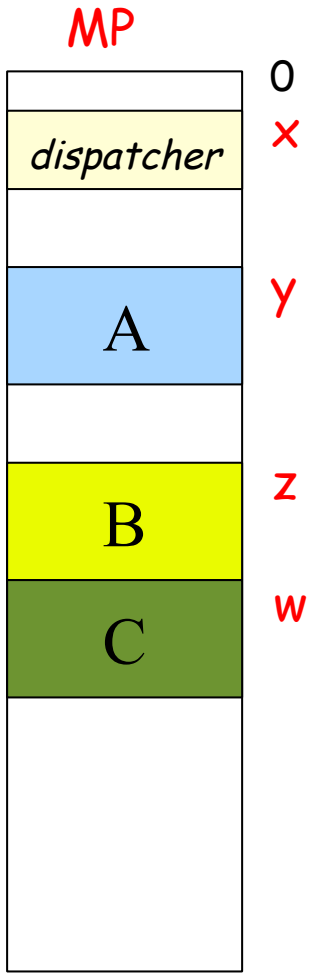
O que esperar do SO?

- Alternar a execução de processos de forma a maximizar a utilização da UCP e fornecer tempo de resposta razoável
- Alocar recursos a processos
- Suportar criação de processos pelo usuário
- Suportar comunicação entre processos

Estados de um Processo

- Processador – sempre executando instruções
  - Definida pelo PC
- PC pode apontar para diferentes processos
- Manipulação realizada pelo *dispatcher (despachante)*
- *Trace de um processo*
  - *Com multiprogramação, traces de vários processos são intercalados*

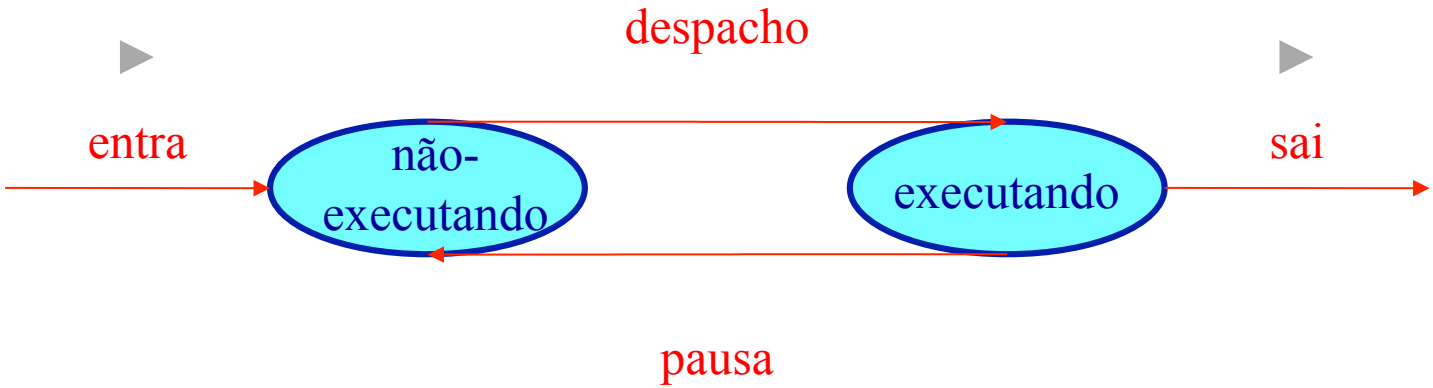
Estados de um Processo



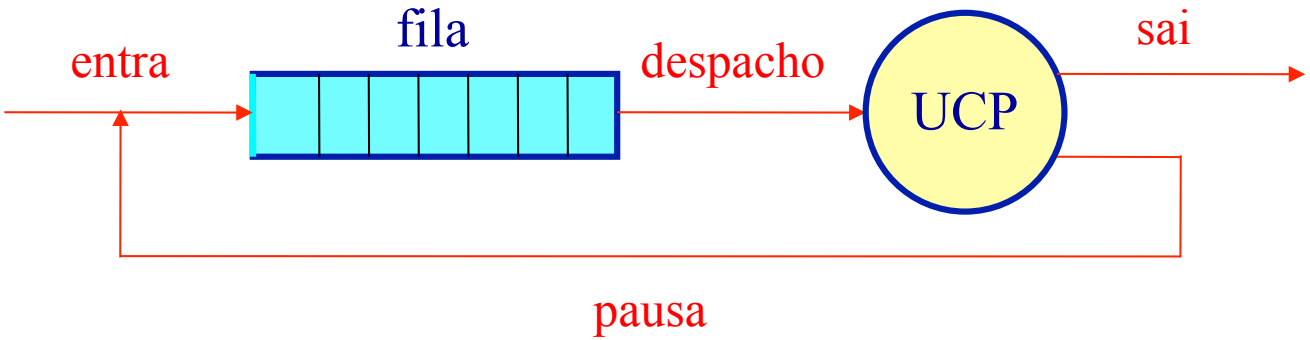
- Cada ciclo = uma instrução
- quantum = 6 ciclos
- Trace:

x, x+1, ..., x+5, y, ...y+5, x, x+1, ..., x+5,
z, ..., z+5, x, x+1, ..., x+5, w, ...w+5,...

Modelo simples de processo



(a) diagrama de transição de estado



(b) possível implementação

Criando de processos

- O que faz o SO para criar processos?
  - constrói estruturas de dados
  - aloca espaço de endereçamento

Criando de processos

- Quando cria?
  - quando usuário abre sessão;
  - quando gerado por outro processo (e.g., servidor de FTP); ...
  - Submissão de um job (batch)
  - Processo cria outros (spawn)
    - Para explorar paralelismo/concorrência

Terminando processos

- Quando terminar?
  - execução da instrução *Halt;*
  - condições de erro; ...

- Algumas razões para o término:
  - tempo excedido;
  - falta de memória;
  - uso de instrução privilegiada;
  - término do processo pai; ...

Problema com o modelo simples

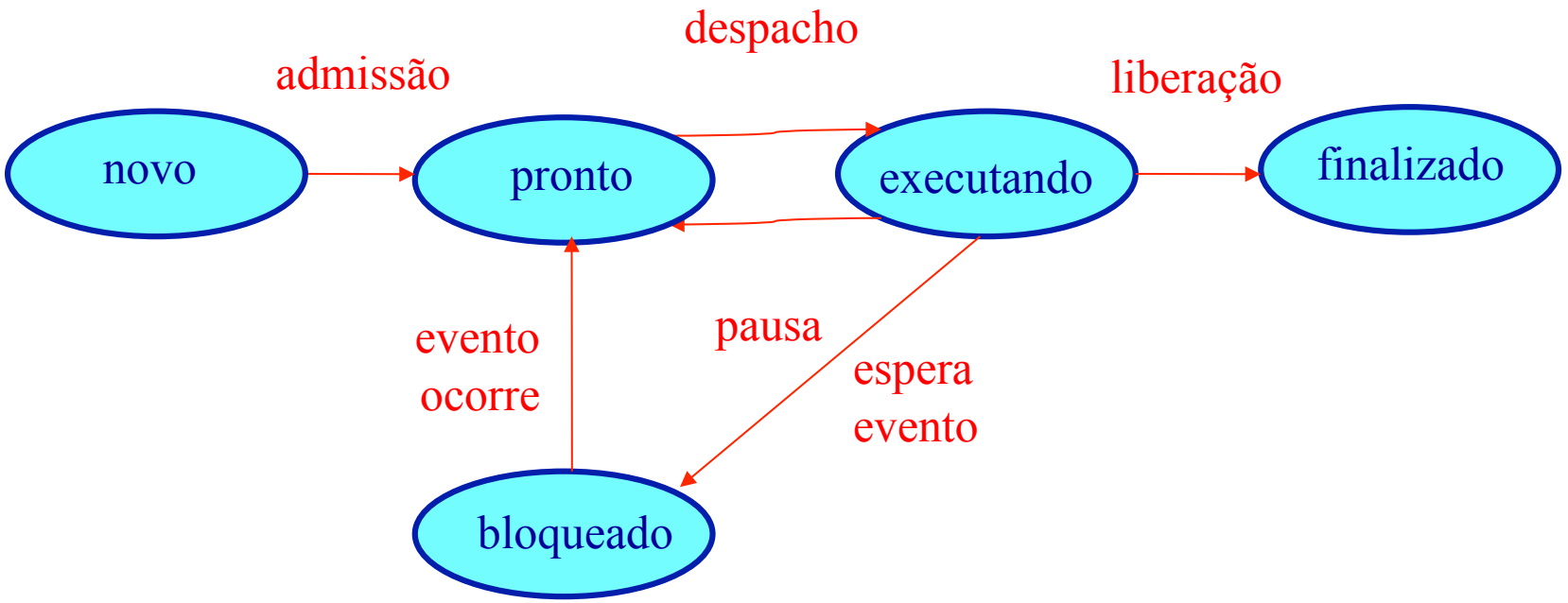
- Um processo que não está executando estará sempre pronto a executar?

- Não:

⇒ pode estar bloqueado esperando E/S!

⇒ despachante não pode simplesmente pegar um processo que esteja na fila

Um modelo mais elaborado – 5 estados



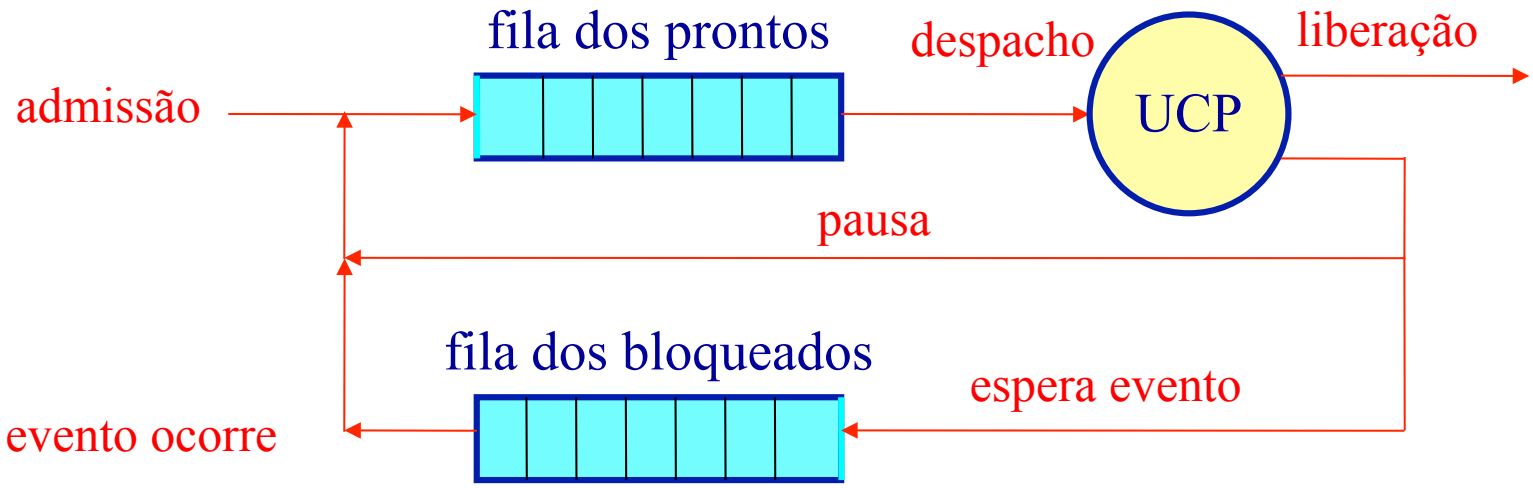
Modelo de 5 estados

- Novo
  - criado, mas não necessariamente admitido no sistema
- Finalizado
  - Término da execução – não mais pronto
  - Ainda com dados para análise de desempenho
- Executando <-> Pronto
  - Time-slice ou por prioridade (neste caso, preempção)

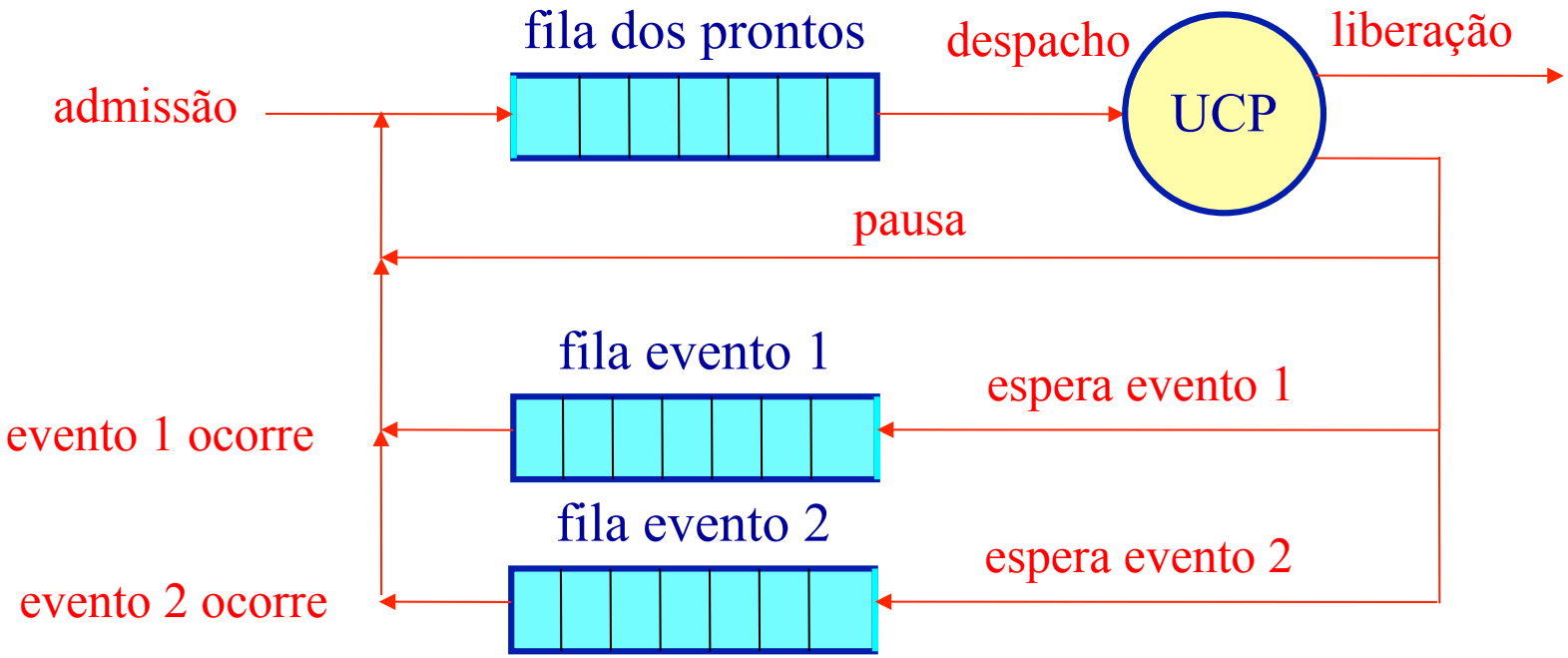
Modelo de 5 estados

- Preempção X não preempção
- Executando <-> bloqueado
  - Chamada ao sistema: E/S, alocação de MP, comunicação entre processos
- Bloqueado <-> pronto
  - Chamada foi atendida

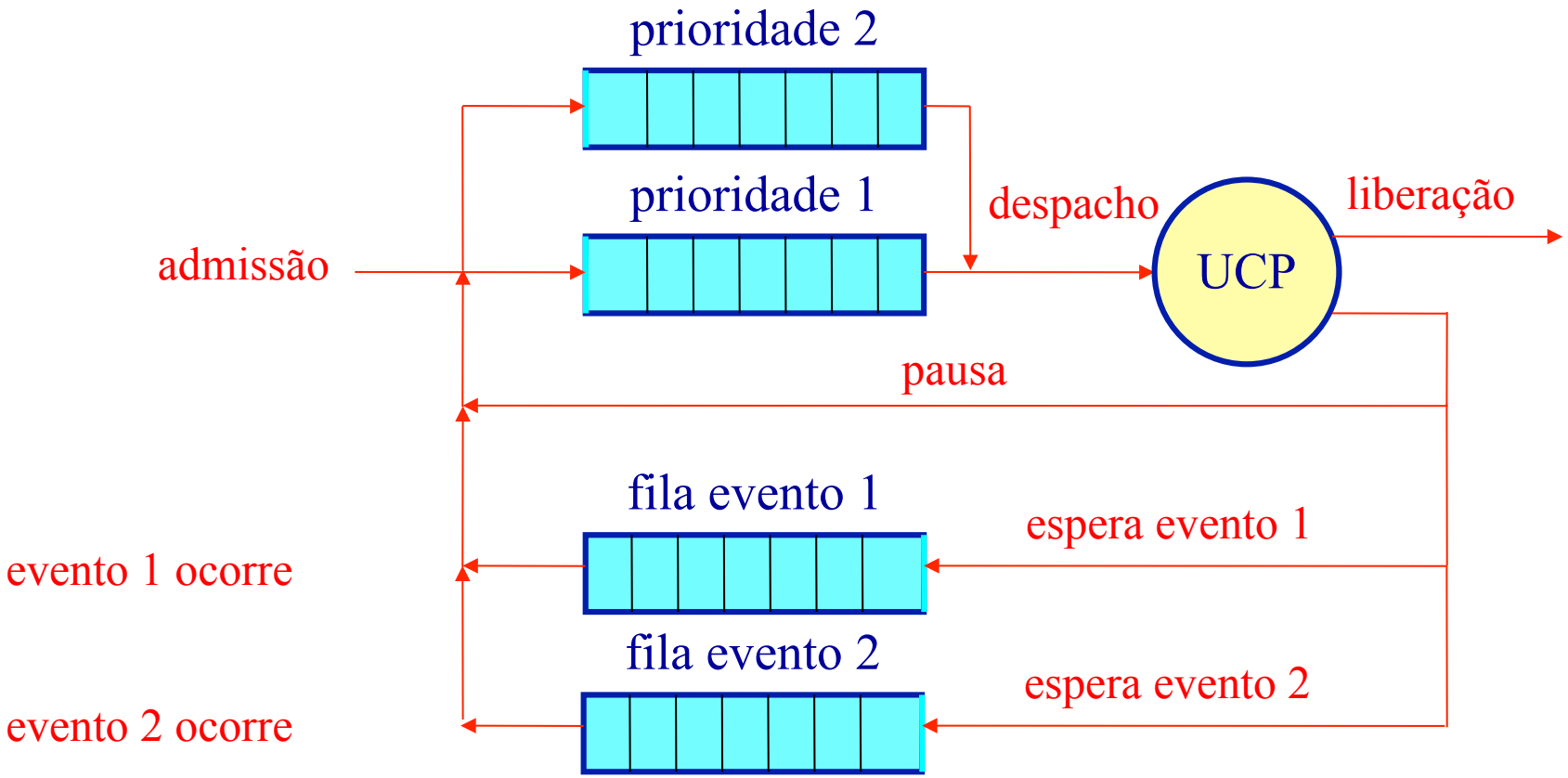
Implementando o modelo (1)



Implementando o modelo (2)



Implementando o modelo (3)



O estado Suspenso

- Vários processos em execução – necessidade de espaço em MP disponível
- Importante para implementação de memória virtual
- O processador é muito mais rápido que E/S: *todos* os processos podem estar bloqueados
  - Ocupação desnecessária de MP

O estado Suspenso

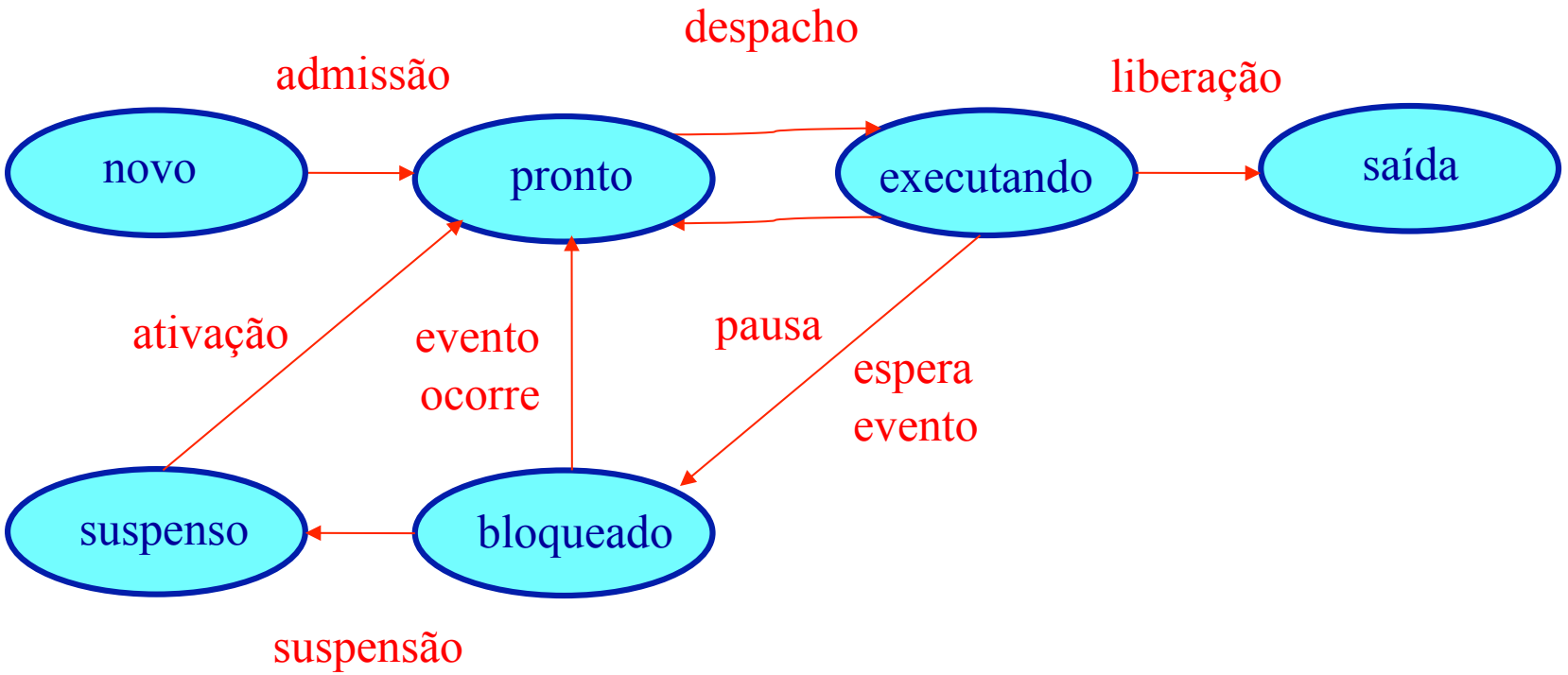
Swapping

- Necessidade de novo estado: representação de processo suspenso
  - Imagem do processo sai temporariamente da MP
  - Quando nenhum processo em MP está pronto (quase todos bloqueados esperando por eventos)
  - É um operação de E/S

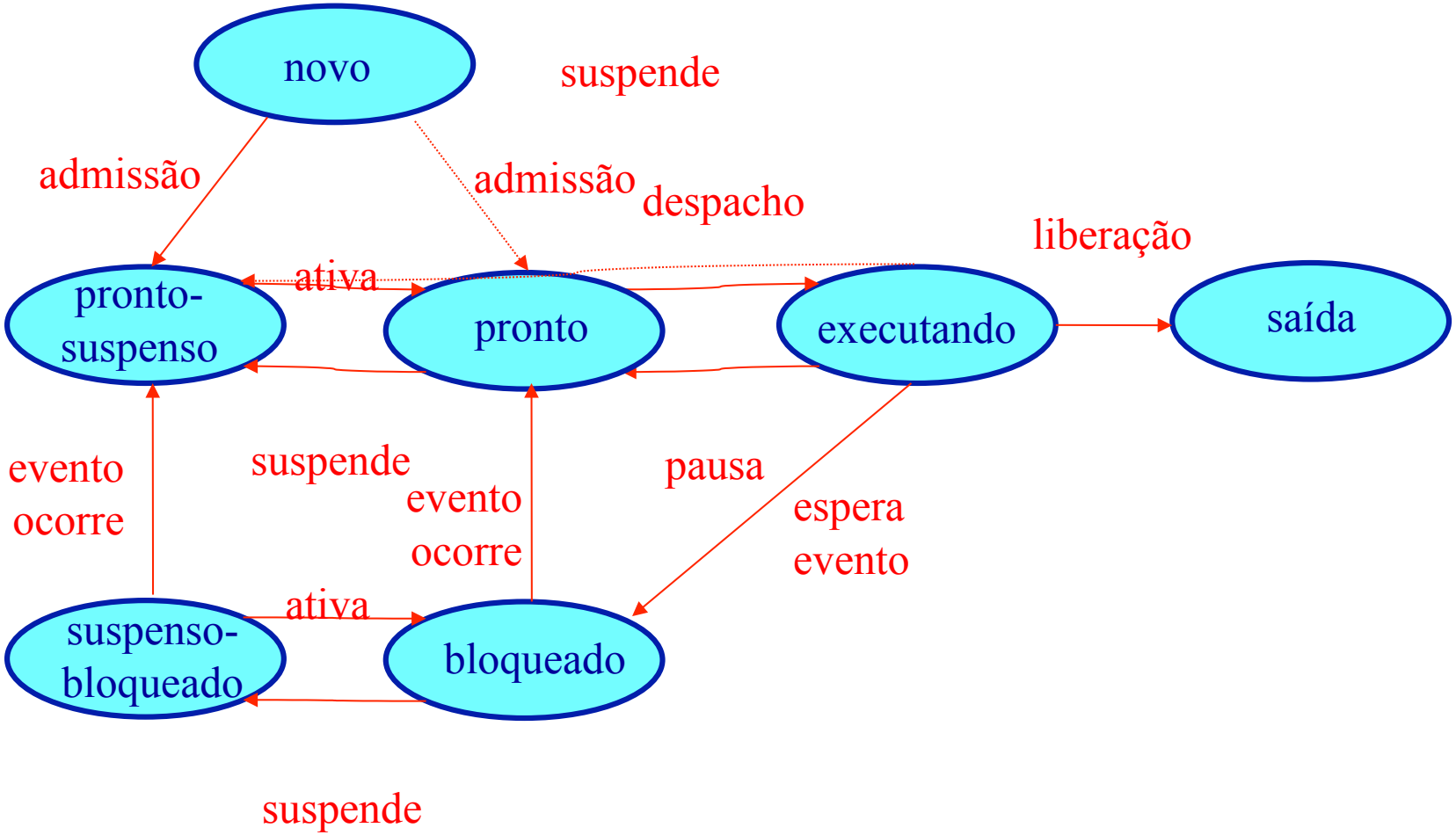
Swapping-out

- *SO seleciona um dos bloqueados para sair de MP*
- *SO seleciona um processo para MP*

Diagrama com estado Suspenso



Pensando melhor ...



O que o SO deve fazer?

- Em um sistema multi-tarefas, o SO deverá:
  - escalonar e despachar processos para execução
  - alocar recursos aos processos
  - responder a pedidos de recursos feitos pelos programas dos usuários
- Como fazer?

Tabelas!

- Tabelas de memória
  - Info sobre alocação de MP (e MS, caso MV seja implementada) aos processos
  - Atributos de proteção, regiões compartilhadas
- Tabelas de E/S
  - Para controle de dispositivos e canais do sistema
  - estado de dispositivo de E/S

Tabelas!

- Tabelas de arquivos
  - localização de arquivos em MS
- Tabelas de processos
  - lista de processos existentes

- As tabelas estão conectadas entre si
- São utilizadas por diferentes gerenciadores (módulos do SO)

Controlando processos

- Para gerenciar processos o SO precisa conhecer:
  - onde o processo está localizado
  - os atributos do processo
- Como é representado um processo?
  - imagem do processo: programa + dados + pilha(s) + atributos
    - atributos = info necessárias pelo SO
    - imagem está na MP

Atributos do processo

- O conjunto dos atributos de um processo é conhecido como *bloco de controle do processo* (PCB, em inglês)
- PCB pode ser dividido em três partes:
  - identificação do processo
  - informação de estado do processador
  - informação de controle do processo

Identificação do processo

- Feita por identificadores numéricos que incluem:
  - identificador do processo
    - em referência cruzada
    - Tabelas
    - Comunicação entre processos
  - identificador do processo que o criou (pai)
  - identificador do usuário (que é um processo)

Informação do processador (estado do processo)

- Contida nos registradores do processador:
  - registradores visíveis ao usuário (aqueles referenciados por programa)
  - registradores de controle e estado (e.g., PC, IR, SO, PSW (códigos de condição))
  - apontadores de pilha

Informação de controle

- Estado e escalonamento, que inclue:
  - estado do processo (e.g., pronto)
  - nível corrente de prioridade
  - suporte ao escalonamento (e.g., há quanto tempo o processo está esperando)
  - evento (e.g., identificação do evento que o processo está esperando)

Informação de controle

- Estruturação de dados (e.g., relação pai-filho)
- Comunicação entre processos (e.g., sinais)
- Privilégios (e.g., tipos de instruções que podem ser executadas)
- Gerenciamento de memória (e.g., ponteiro para tabela de páginas)
- Propriedade e uso de recursos (e.g., arquivos abertos)

Controle de Processos

- Ao longo da execução de processos, da utilização do sistema, passos importantes devem ser executados para controlar a execução de processos
  - modos de execução
  - criação de processos
  - chaveamento entre processos
  - mudança do estado do processo

Modos de execução de processador

- Se o sistema só executa um processo por vez
  - qualquer problema causado pelo processo só causaria problema ao próprio
- No entanto, com o compartilhamento dos recursos aos vários processos
  - proteção deve ser cuidadosamente manipulada

Modos de execução de processador

- Proteção necessária em um ambiente multiprogramável
  - um programa com erro pode modificar outro programa, dados de outro programa, até o próprio núcleo do SO
  - Execução de instruções privilegiadas por parte de processos de forma descuidada

Como resolver?

Modos de execução de processador

- Muitos SOs definem diversos modos de execução, através de suporte de hardware para implementar proteção

Modos de execução de processador

- Modo dual de operação: um bit determina em qual modo de operação o sistema está executando
  - Modo privilegiado/supervisor/núcleo
    - certas funções são somente executadas em modo privilegiado
  - Modo do usuário
    - modo de execução dos processos comuns

Modos de execução de processador

- Ao ligar o computador, no momento do *boot* do sistema
  - Hardware começa em modo supervisor
  - SO é carregado
  - passa para modo usuário para iniciar processo

Modos de execução de processador

- Sempre que executa um processo, o sistema está em modo usuário (bit = 1)
- Passa para modo supervisor (bit = 0, setado pelo hardware)
  - *trap*
  - interrupção
  - chamada ao SO
    - na verdade, sempre que SO obtém controle para gerenciar
      - processos, memória, E/S e outras funções de suporte como interrupção, monitoramento e contabilidade

Modos de execução de processador

- Ao voltar a executar um processo de usuário, o SO passa para o modo usuário (bit = 1)

Modos de execução de processador

- O hardware permite que instruções privilegiadas sejam executadas somente em modo supervisor
  - desta forma, o sistema está protegido contra usuários mal intencionados ou descuidosos
- Instruções privilegiadas
  - definidas naquela arquitetura
  - programa do usuário faz chamadas ao SO: execução de instruções privilegiadas

Modos de execução de processador

- Especificação do modo de execução do processador
  - tipicamente, pode ser usado um bit de modo do PSW do processo em questão

Criação de processos

- atribuição de identificador único ao processo
- adição de nova linha à tabela de processos
- alocação de espaço para a imagem
  - código+dados+pilha+bloco de controle
- iniciação do PCB
  - inclui identificação nas listas apropriadas (e.g., lista de prontos para escalonamento)

Chaveamento entre processos

- interrupções
  - relógio (mudança de estado, escalonamento)
  - E/S
  - falta de memória (página ou segmento)
    - o processo fica bloqueado para leitura de disco
- *traps*: condições anormais
- chamada ao sistema (e.g., operação de E/S)

Chaveamento entre processos

Tratamento de interrupção

- Processador fica em modo *kernel/núcleo*
  - informações de controle salvas no PCB
  - seta o PC para endereço da rotina do SO de tratamento de interrupção
  - despachante pode escalonar outro processo depois do tratamento
    - ou, no caso de interrupção por E/S, o SO pode escolher o processo interrompido para continuar a ser executado e economizar tempo na troca de contexto

Muito é investido no hardware para minimizar o custo de tratamento de interrupção (salvamento de contexto)

Mudança de Estado de Processo

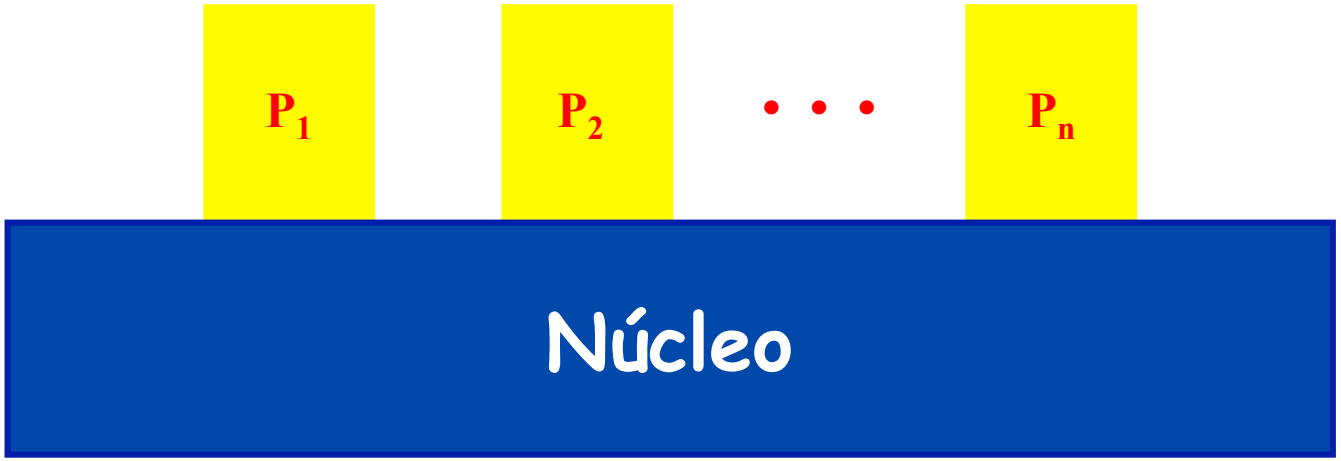
- Sobrecarga associada à mudança de estado realizada pelo SO
  - salva contexto do processo
  - atualiza bloco de controle do processo (PCB)
    - gravação do novo estado (pronto/bloqueado/ suspenso...)
    - gravação do motivo da mudança de estado
  - move o processo (PCB) para a fila apropriada
  - escolhe novo processo para execução
  - atualiza PCB do novo processo e dados relativos a MP
  - restaura contexto do novo processo

Executando o SO

- O SO é um programa. Ele é um processo?
  - depende de como o projetista implementa o SO
  - funções executadas como processo de usuário: só há chaveamento de modo
  - SO como conjunto de processos de sistema: funções principais do núcleo como processos

Núcleo separado

- núcleo fora dos processos, alocado em uma determinada região da MP
- salva contexto, escalona e despacha processos....
- o processo é interrompido para o SO entrar em ação



Funções em modo usuário

- comum em microcomputadores: as funções chamadas no processo estão embutidas na imagem deste
- código e dados compartilhados entre o processo e SO
- quando interrupção, o próprio processo é colocado em modo privilegiado
  - troca de modo é executado e não a troca de processo

Como processos de sistema

- SO em módulos – pequenos processos SO (e.g., serviços)
- troca de contexto menos custosa
- mais vantajoso para multicomputadores

Exercício I – para entregar

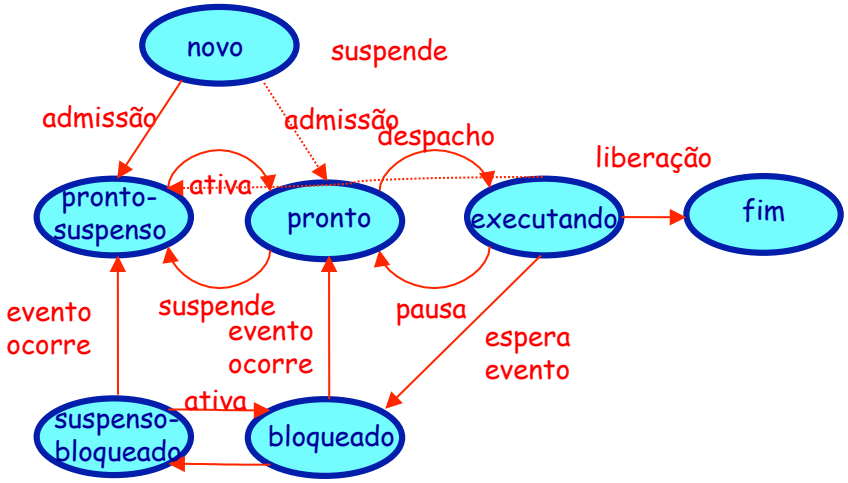
Considere o diagrama de 7 estados. Suponha que esteja no momento do SO escalonar um processo e existem processos tanto em estado pronto, quanto pronto-suspenso. Pelo menos um processo em pronto-suspenso tem prioridade maior do que um processo pronto. Duas políticas de escalonamento são as seguintes:

1) sempre escalonar um processo da fila de prontos

2) sempre escolher processo de maior prioridade

Quais as vantagens e desvantagens de cada uma das duas políticas? (dê exemplos)

Sugira uma política intermediária?



Exercício II – para entregar

Para que serve o bloco de controle do processo (PCB)?

Quais as informações armazenadas e qual a justificativa para cada uma das informações?