

Sistemas Operativos

Licenciatura em Engenharia Informática
Licenciatura em Engenharia Computacional
Licenciatura em Física (opcional)

Ano letivo 2025/2026

Pedro Azevedo Fernandes (paf@ua.pt)

Processos

Um processo é um programa em execução

- Assume-se um SO multiprogramado que pode comutar entre um processo e outro
- Algumas vezes este mecanismo é designado por pseudoparalelismo...
- ...em contraponto ao paralelismo real, que envolve vários processadores
- O foco deste estudo não envolve paralelismo real

Elementos de um Processo

Um processo possui dois elementos essenciais:

Código de programa (program code)

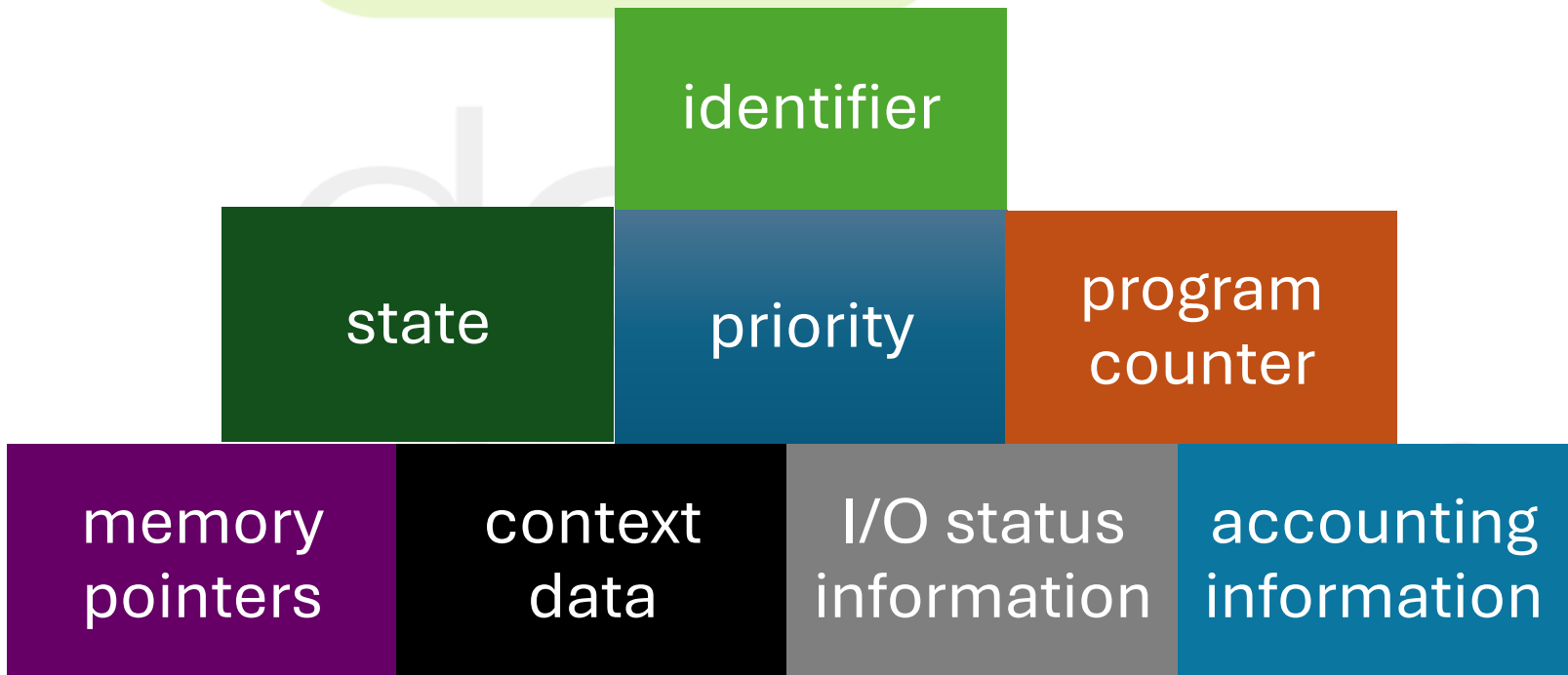
que pode ser partilhado por outros processos que estejam a correr o mesmo programa

Conjunto de dados associado a esse código

Quando o processador inicia a execução do código de programa, essa entidade designa-se por **processo**

Elementos de um Processo

- Enquanto um programa está em execução, esse processo pode ser caracterizado de forma única por um conjunto de elementos, incluindo:



Bloco de Controlo de Processos (PCB) - 1

O PCB:

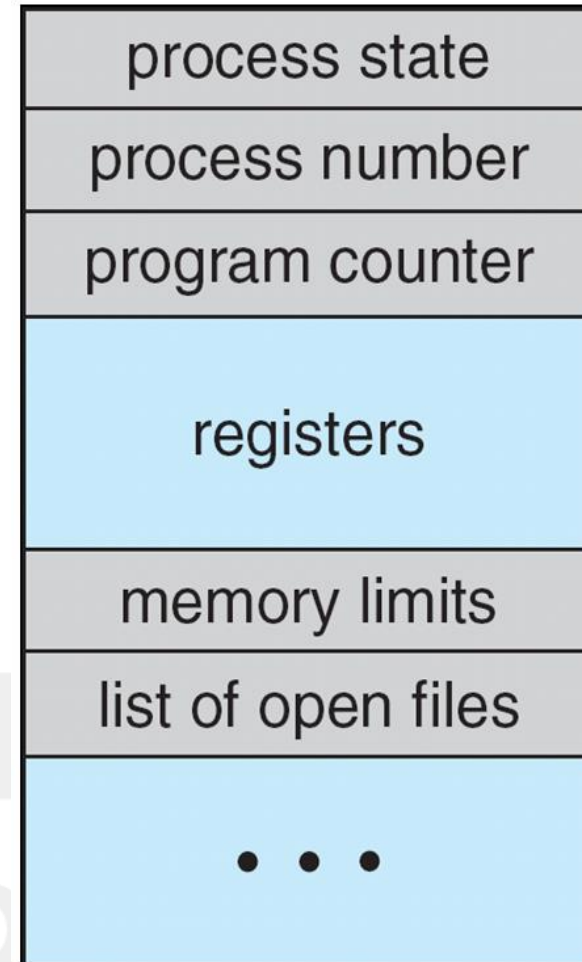
- contém os elementos de um processo;
- é possível interromper um processo e, mais tarde, retomar a sua execução, como se nada se tivesse passado
- é criado e gerido pelo SO
- é uma peça chave que permite que múltiplos processos possam coexistir

Identifier
State
Priority
Program counter
Memory pointers
Context data
I/O status information
Accounting information
⋮

Bloco de Controlo de Processos (PCB) - 2

Informação associada com cada processo:

- Process state – correr, preparado, ...
- Program counter – localização da próxima instrução a se reexecutada
- Registos da CPU – conteúdo de todos os registos relacionados
- Scheduling info da CPU – prioridades, apontadores de scheduling, ...
- Informação de gestão de memória – memória atribuída ao processo
- Informação de contabilização – CPU utilizada, tempo de relógio desde o início, ...
- Estado das E/S – Dispositivos E/S atribuídos ao processo, ficheiros abertos, ...



Bloco de Controlo de Processos (PCB) - 3

O PCB:

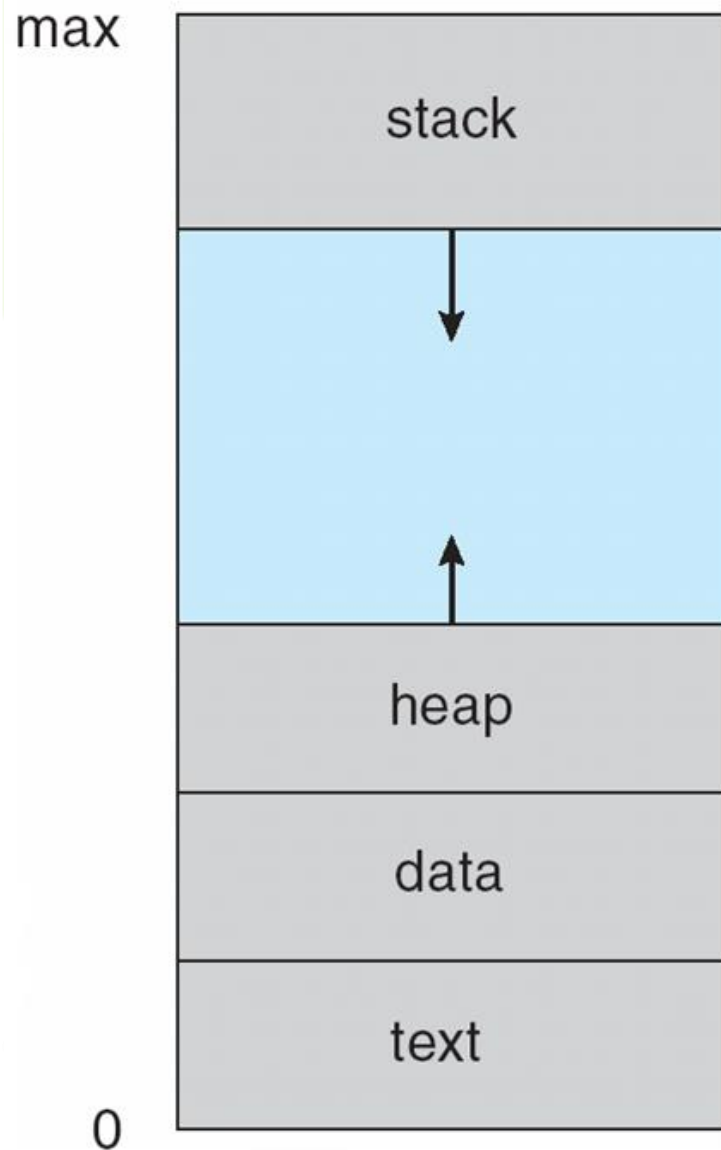
- é a estrutura de dados mais importante de um SO
 - contém toda a informação acerca de um processo que o SO necessita
 - os PCB são de leitura e/ou escrita por praticamente todos os módulos do SO
 - define o estado do SO
- a dificuldade não é aceder-lhes, mas protegê-los
 - um bug pode danificar a sua estrutura, o que pode impedir a gestão do respetivo processo
 - uma alteração da estrutura do PCB pode afetar diversos componentes do SO

Conceito de Processo

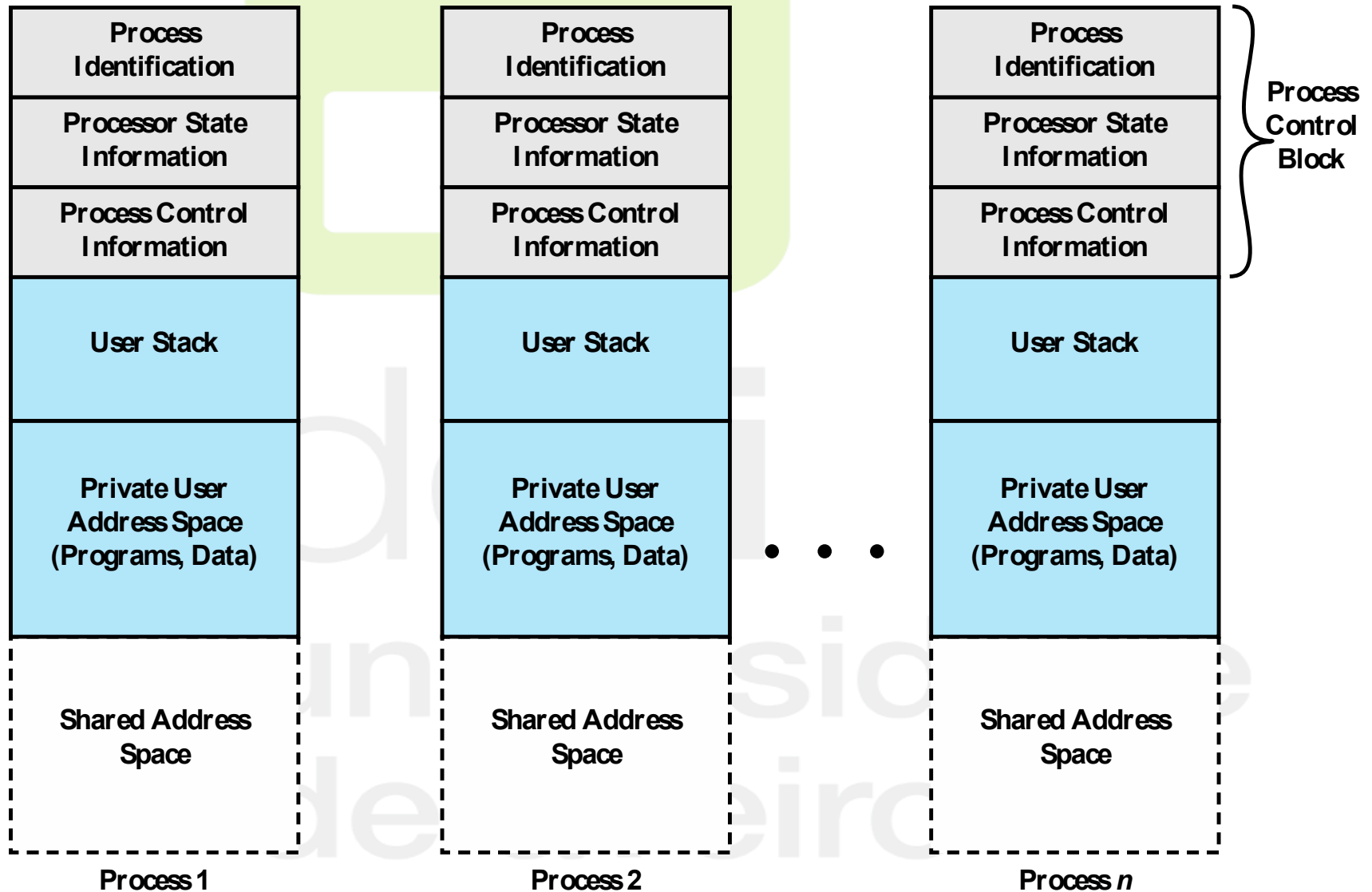
Um processo é um programa em execução e inclui:

- O código de programa (program code) também é designado por **text section**
- A atividade corrente, incluindo o **program counter**, processador, registos
- **Stack** contendo dados temporários
 - Parâmetros de funções, endereços de retorno, variáveis locais
- **Data section** contendo variáveis globais
- **Heap** contendo memória dinamicamente atribuída durante a sua execução

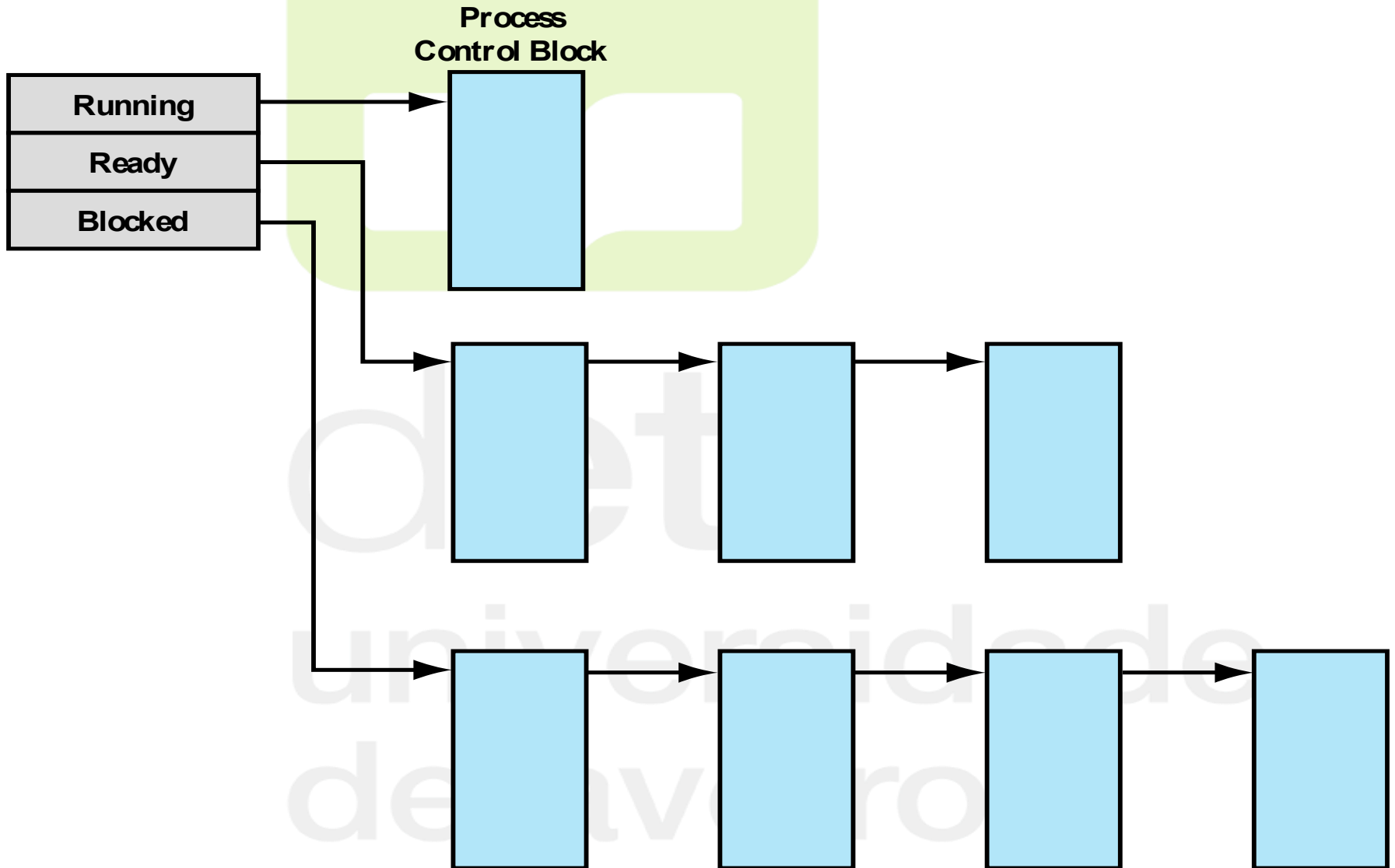
Processo na Memória



Processos na Memória Virtual

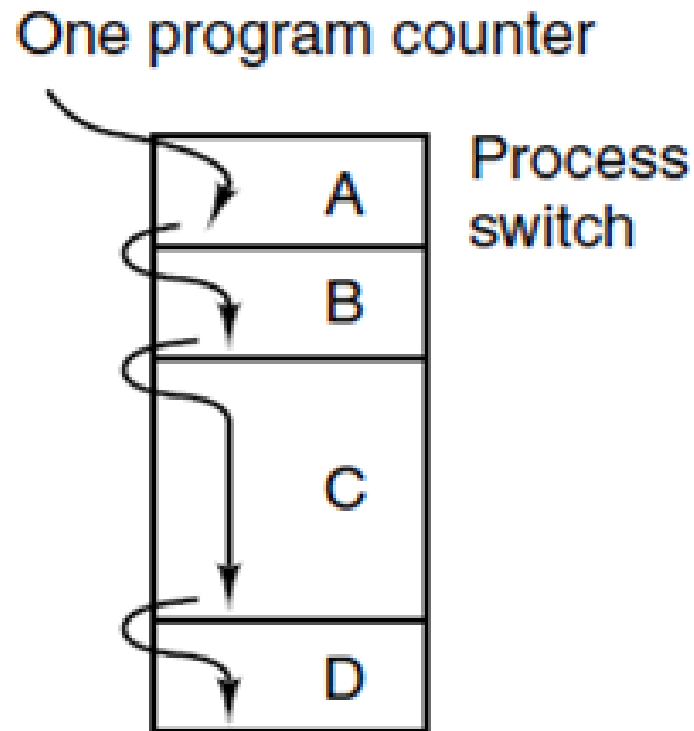


Estrutura de Listas de Processos



O Modelo de Processos – 1

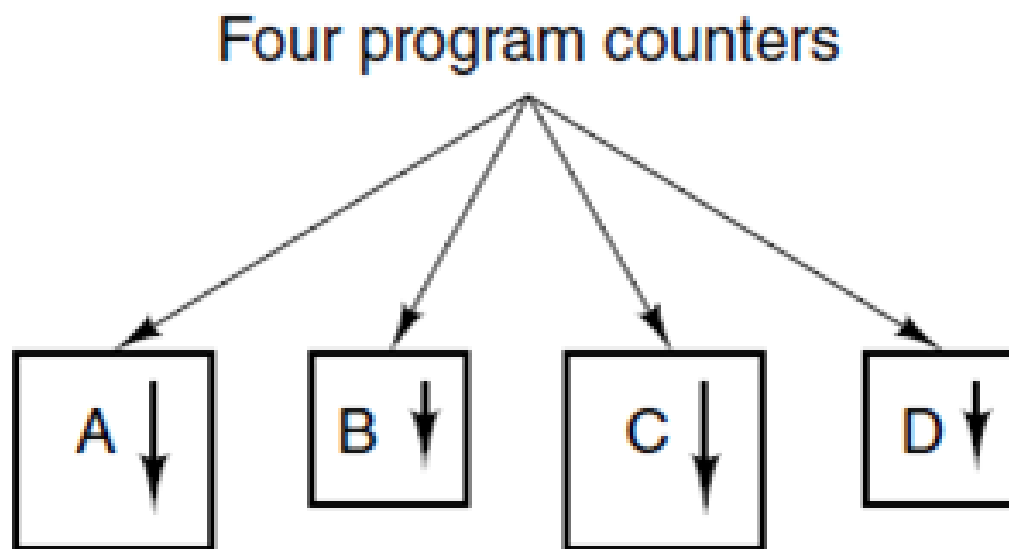
- Multiprogramação de quatro programas



(a)

O Modelo de Processos – 2

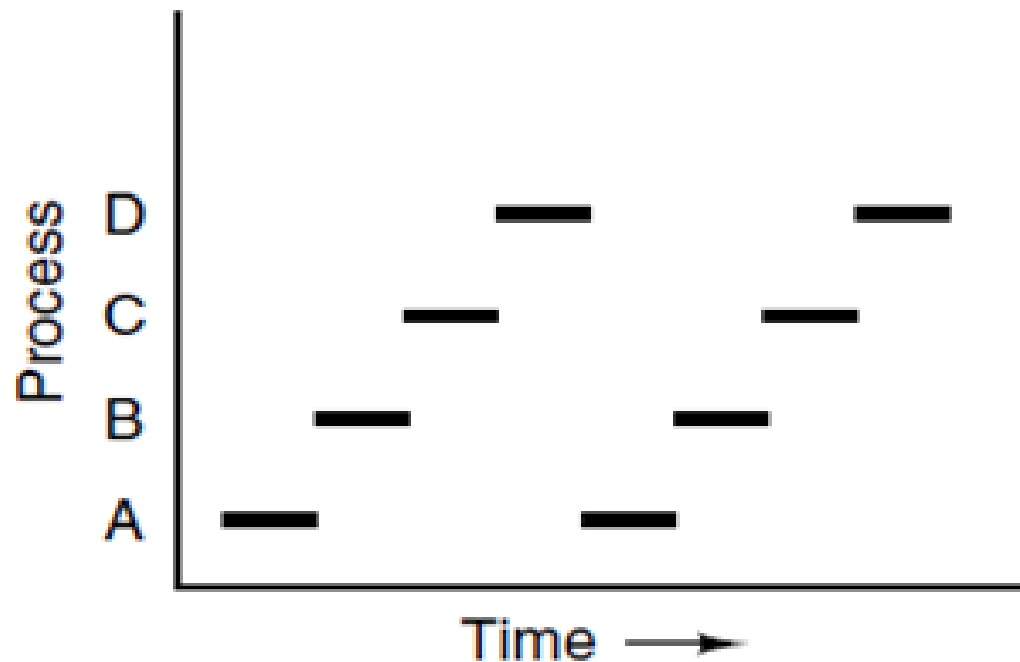
- Quatro contadores de programas



(b)

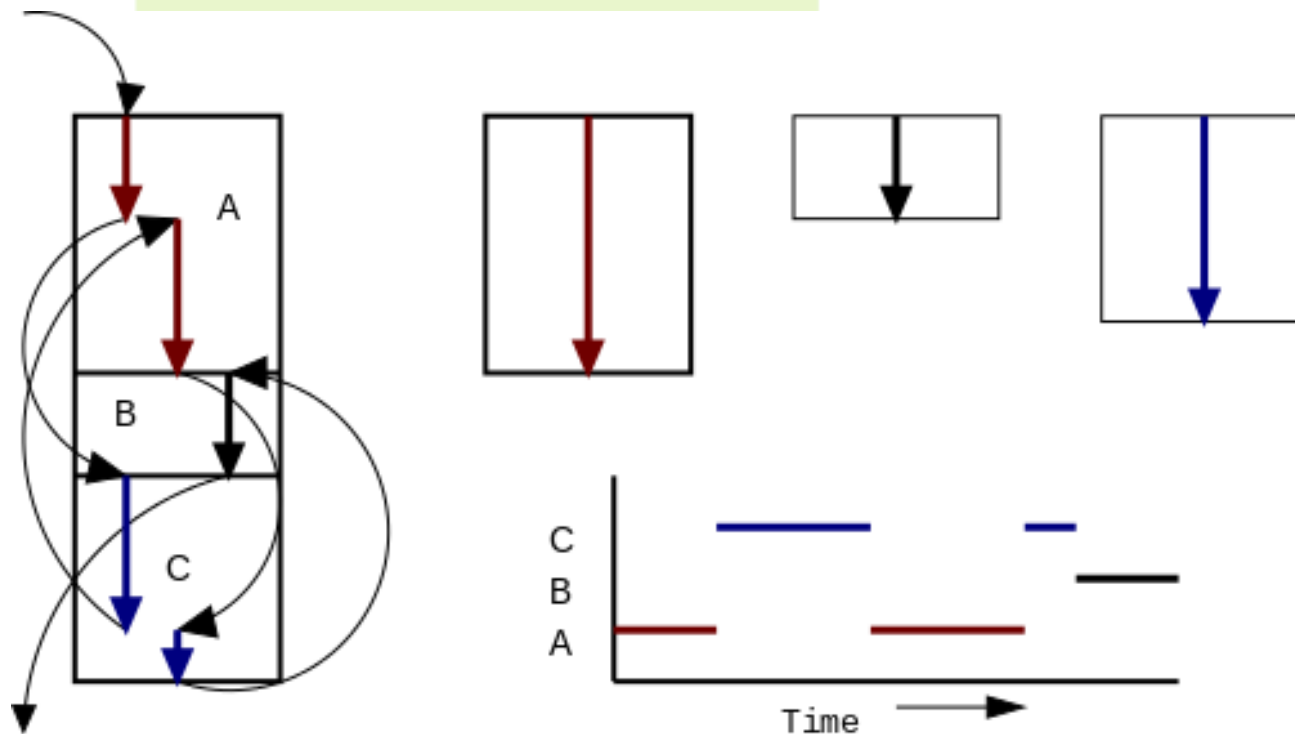
O Modelo de Processos – 3

- Apenas um programa está ativo a dado momento



(c)

O Modelo de Processos - 4



Three views of multiprogramming with three processes:
The real program counter; each virtual program counter;
time history of process execution. (We assume no jumps.)

O Modelo de Processos – considerações

- Apesar da existência de vários processos a correr, cada um tem a ilusão de correr sozinho
- Tempo virtual
 - É usado apenas por cada processo em causa
 - O tempo virtual progride a uma cadência “independente” dos outros processos
- Memória virtual
 - Memória vista pelo processo
 - Cada processo “acredita” que possui um bloco de memória a começar no endereço 0 (zero)

Modos de Execução de Processos

- User Mode
 - Modo com menores privilégios
 - programas de utilizador executam habitualmente neste modo
- System Mode
 - Modo privilegiado
 - Também referido como control mode, ou kernel mode
 - kernel do sistema operativo

Criação de Processos - 1

Existem quatro situações que levam à criação de processos:

1. Inicialização do sistema
2. Execução, por parte de um processo a correr, de uma chamada ao sistema para criação de um processo
3. Um utilizador pede para ser criado um novo processo
4. Inicialização de um trabalho *batch*

Criação de Processos - 2

Process spawning

- Quando o OS cria um processo devido a um pedido explícito de outro processo

Parent process

- É o processo criador, o processo original

Child process

- É o novo processo

Criação de Processos - 3

Quando o SO decide criar um processo:

Atribui um identificador de processo
PID único ao novo processo



Reserva espaço para o processo



Inicializa o PCB (bloco de controlo de processo)



Define as linkages adequadas



Cria ou expande outras estruturas de dados

Término de Processos - 1

Algumas situações típicas que levam ao término de processos:

1. Saída normal (voluntário)
2. Saída por erro (voluntário)
3. Erro fatal (involuntário)
4. Abatido por outro processo (involuntário)

Término de Processos - 2

Razões para o término de processos



Normal completion	The process executes an OS service call to indicate that it has completed running.
Time limit exceeded	The process has run longer than the specified total time limit. There are a number of possibilities for the type of time that is measured. These include total elapsed time ("wall clock time"), amount of time spent executing, and, in the case of an interactive process, the amount of time since the user last provided any input.
Memory unavailable	The process requires more memory than the system can provide.
Bounds violation	The process tries to access a memory location that it is not allowed to access.
Protection error	The process attempts to use a resource such as a file that it is not allowed to use, or it tries to use it in an improper fashion, such as writing to a read-only file.
Arithmetic error	The process tries a prohibited computation, such as division by zero, or tries to store numbers larger than the hardware can accommodate.
Time overrun	The process has waited longer than a specified maximum for a certain event to occur.
I/O failure	An error occurs during input or output, such as inability to find a file, failure to read or write after a specified maximum number of tries (when, for example, a defective area is encountered on a tape), or invalid operation (such as reading from the line printer).
Invalid instruction	The process attempts to execute a nonexistent instruction (often a result of branching into a data area and attempting to execute the data).
Privileged instruction	The process attempts to use an instruction reserved for the operating system.
Data misuse	A piece of data is of the wrong type or is not initialized.
Operator or OS intervention	For some reason, the operator or the operating system has terminated the process (e.g., if a deadlock exists).
Parent termination	When a parent terminates, the operating system may automatically terminate all of the offspring of that parent.
Parent request	A parent process typically has the authority to terminate any of its offspring.

Interrupções do Sistema

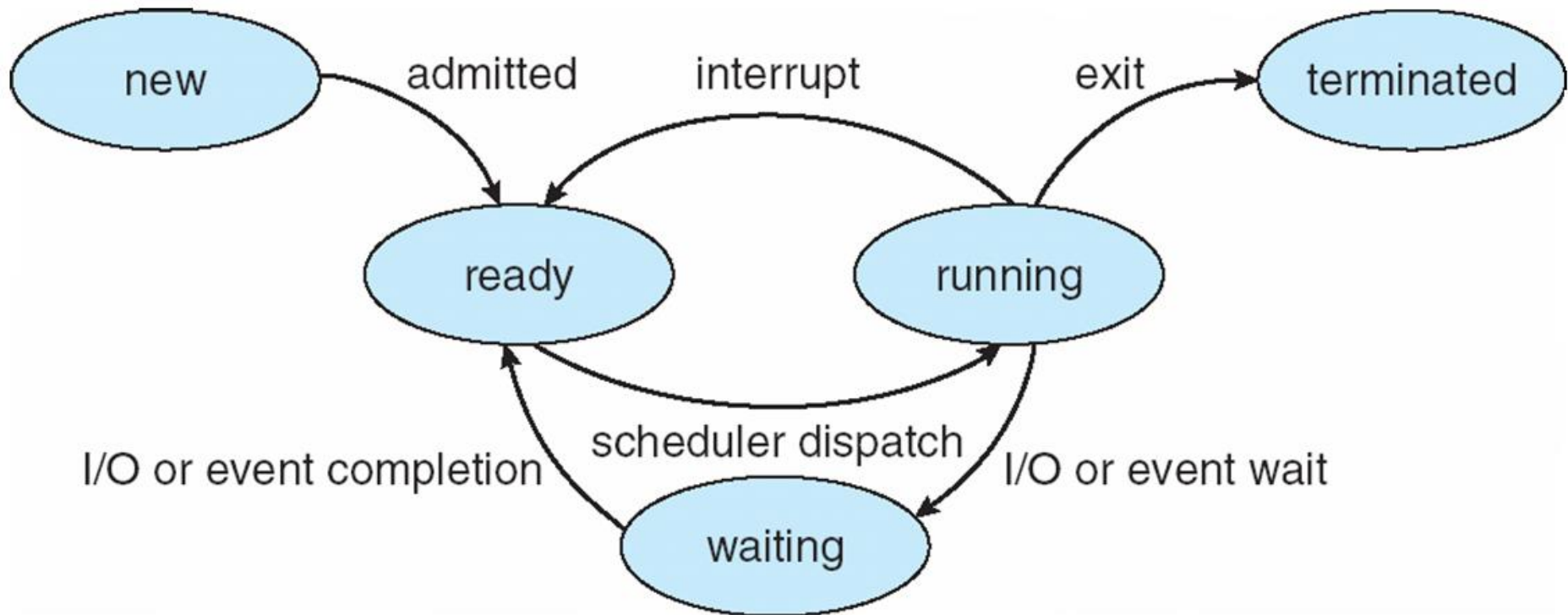
- Interrupção
 - Devido a algum tipo de acontecimento que é externo e independente do processo correntemente em execução
 - Interrupção de relógio
 - Interrupção de E/S
 - Falha da memória
 - Time slice
 - O tempo máximo que o processador pode executar antes de ser interrompido
- Trap
 - Um erro ou condição de exceção gerada dentro do processo correntemente em execução
 - O OS determina se a condição é fatal
 - Movido para o estado Exit (ou Terminated) e ocorre comutação de processos
 - ação dependerá da natureza do erro

Estados de Processos – 1

Quando um processo executa, muda de **estado**:

1. **New**: o processo está a ser criado
2. **Running**: instruções estão a ser executadas
3. **Waiting** (ou **Blocked**): O processo está à espera que algum evento ocorra
4. **Ready**: o processo está a aguardar que seja atribuído a um processador
5. **Terminated** (ou **Exit**): o processo terminou a sua execução

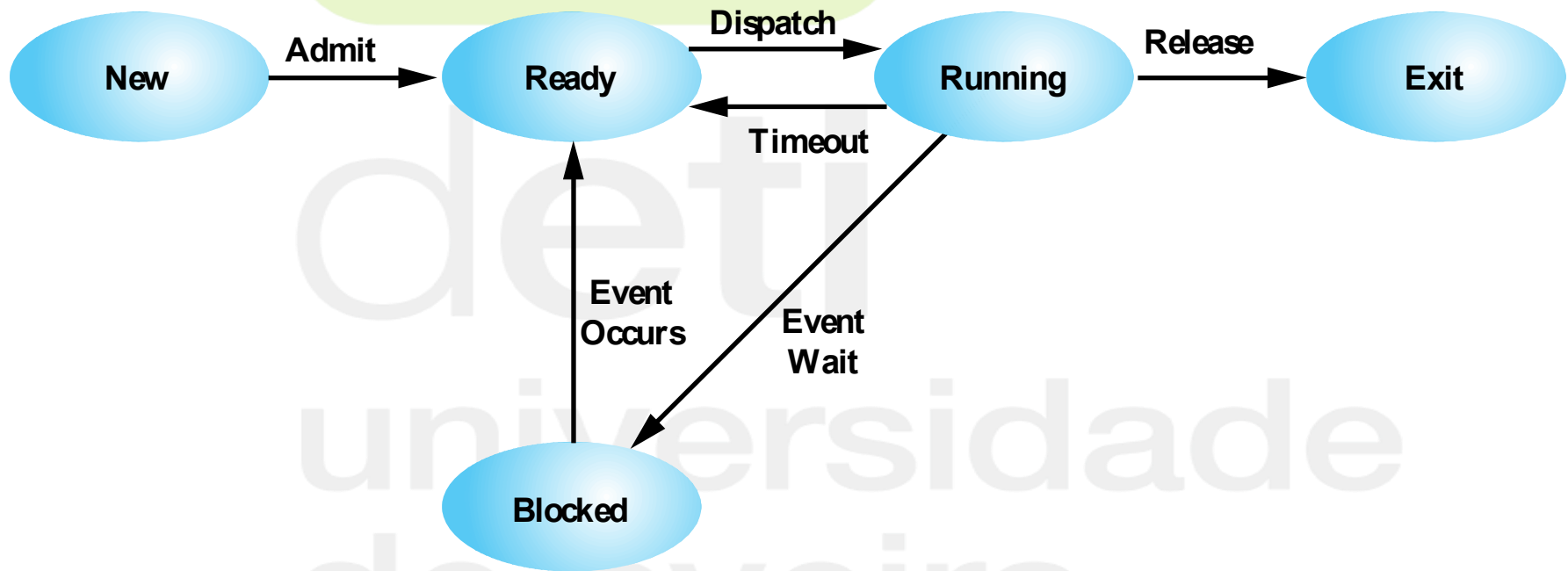
Estados de Processos – 2



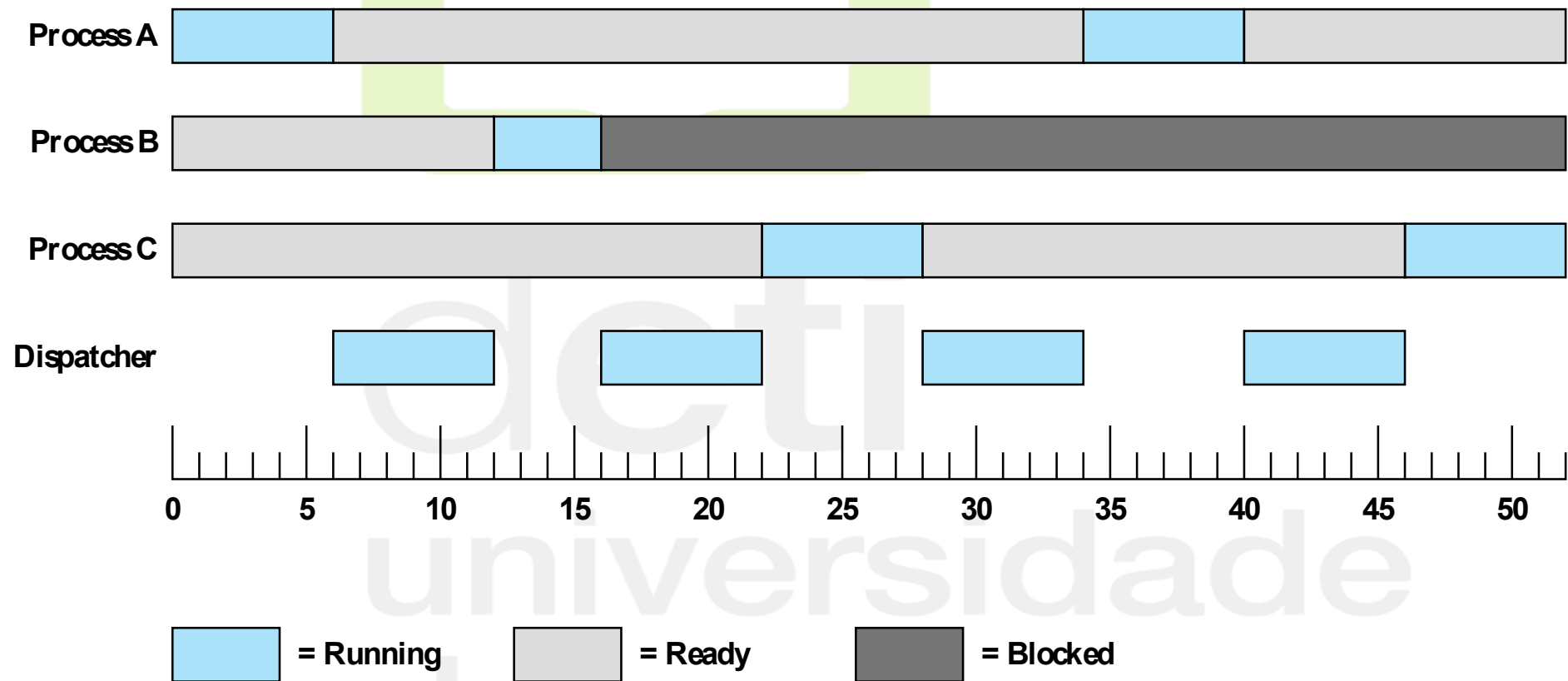
de aveiro

Estados de Processos – 3

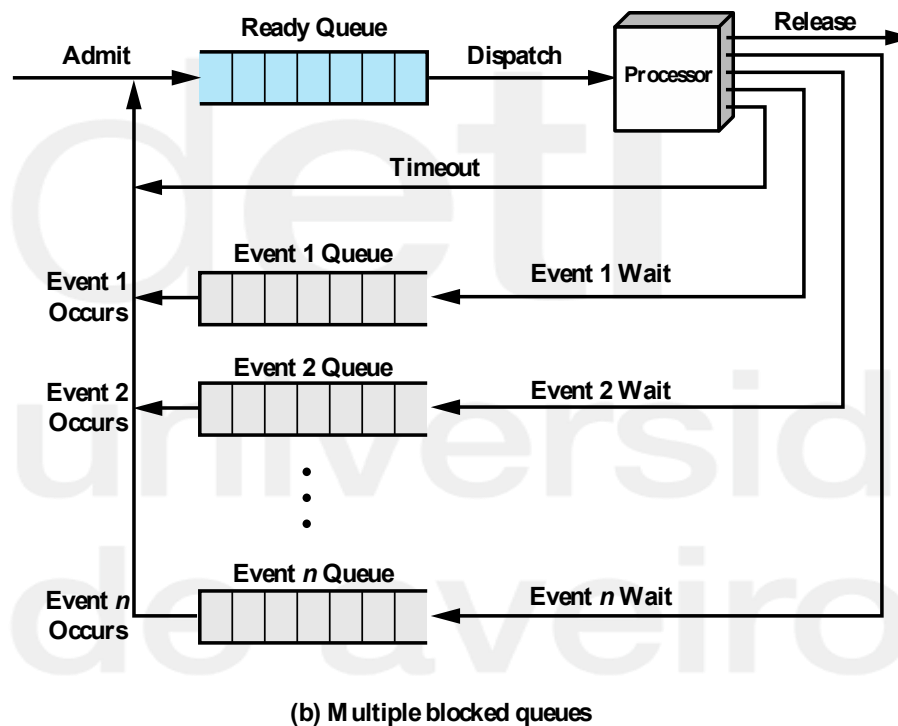
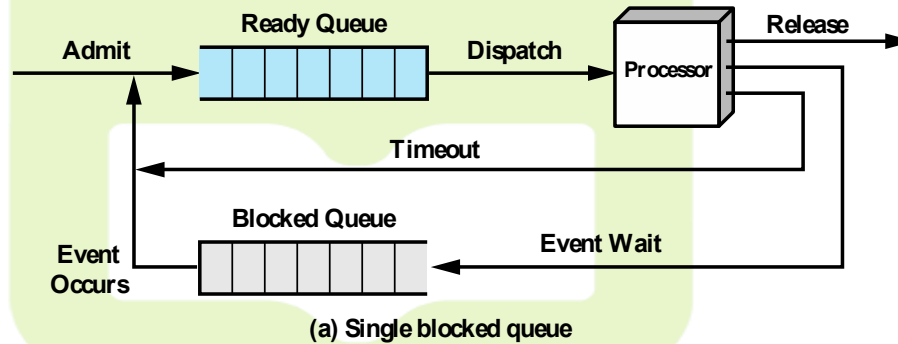
Modelo dos 5 estados



Estados de Processos – 4



Estados de Processos – 5



Estados de Processos – 6

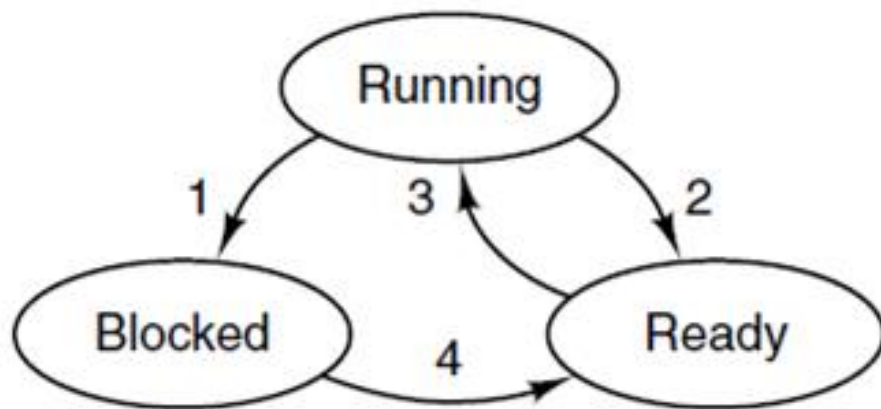
Um processo pode encontrar-se numas das seguintes três situações:

1. A correr (a utilizar a CPU nesse instante)
2. Preparado (pronto para correr; parado temporariamente para deixar outro processo correr)
3. Bloqueado (incapaz de correr até que um acontecimento externo tenha lugar)

Estados de Processos – 7

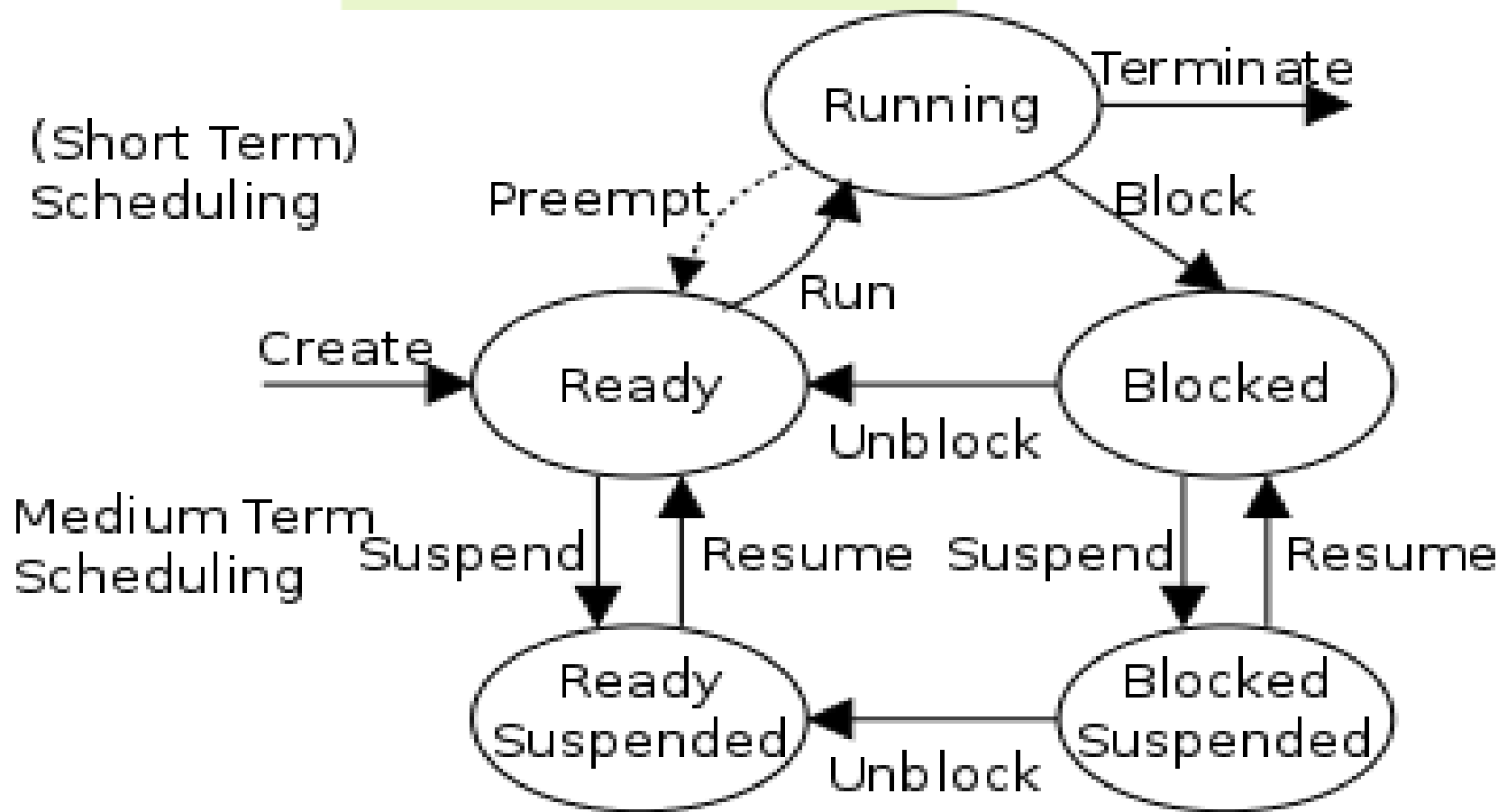
Um processo pode encontrar-se a correr, preparado, ou bloqueado.

Diagrama da transição de estados:



1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

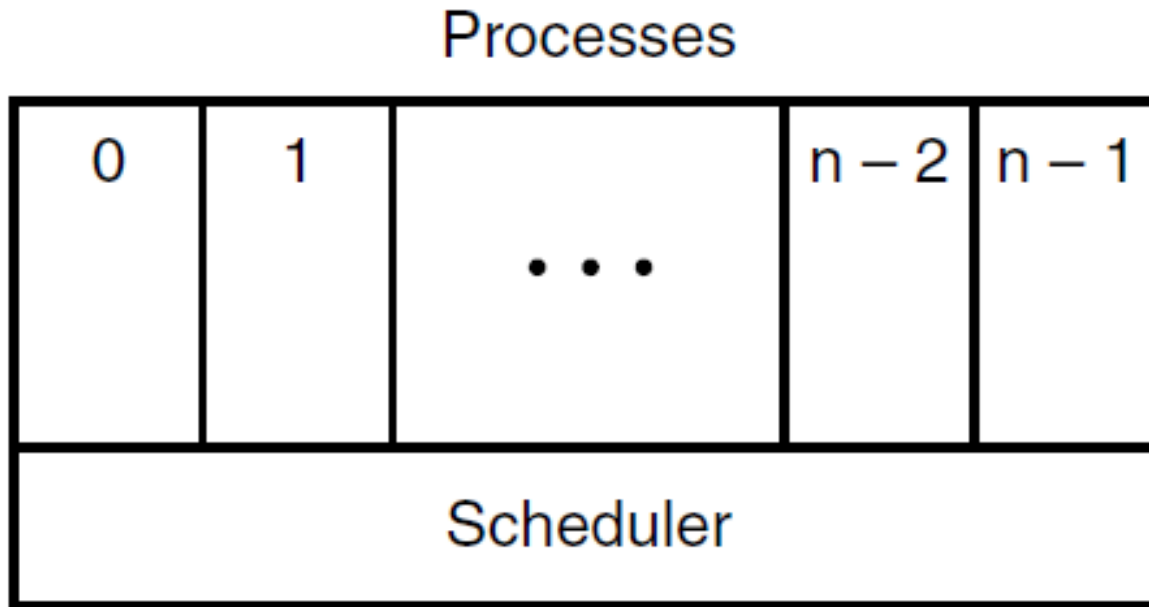
Estados de Processos – 8



Unblock is done by another task (a.k.a. wakeup, release, V)
Block is a.k.a. sleep, request, P)

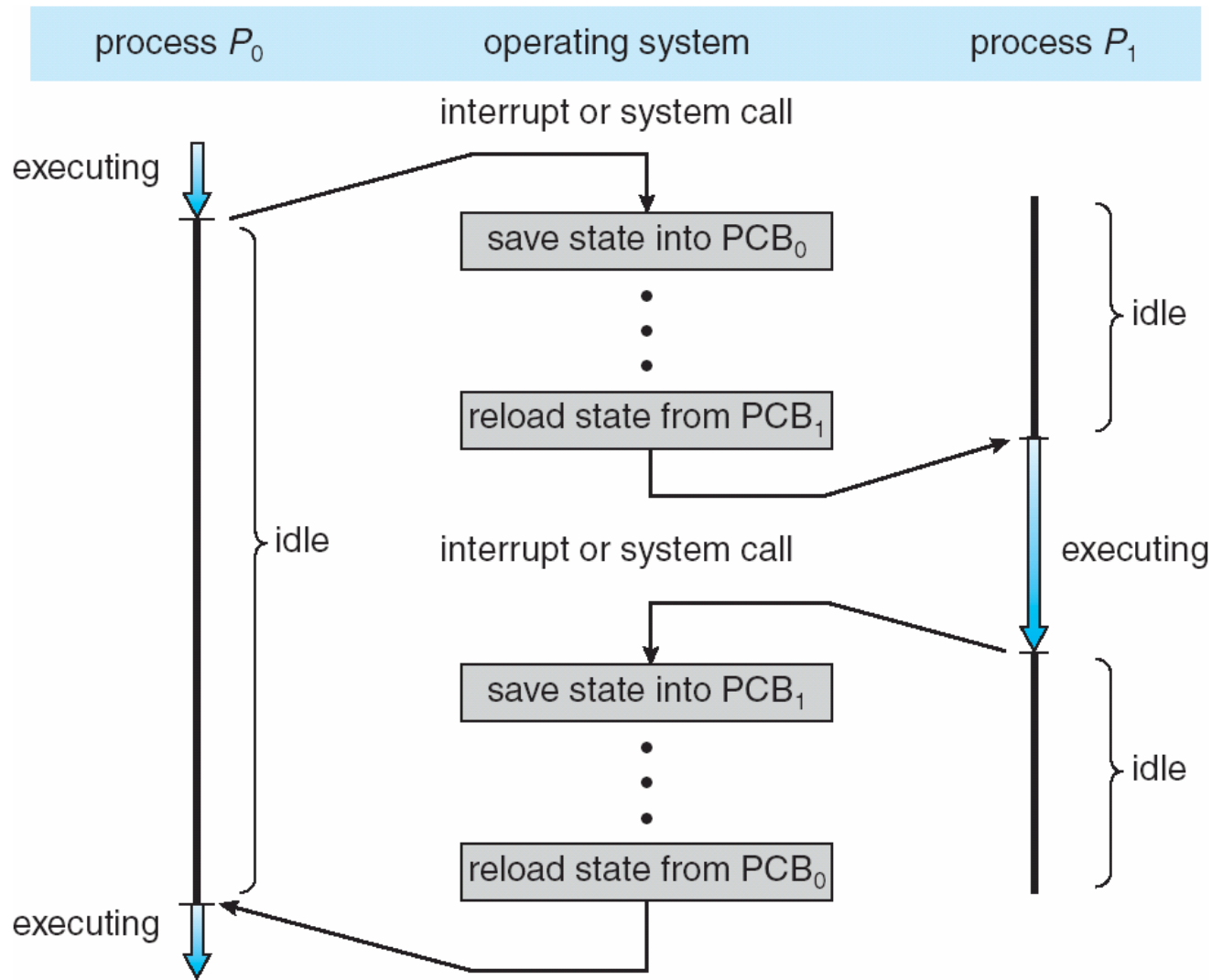
Estados de Processos – 9

As camadas inferiores de um SO estruturado em processos trata das interrupções e scheduling. Os processos correm em cima dessa camada.



ide

Comutação de Processos



Modo de Comutação de Processos - 1

Se não estão pendentes interrupções no processador:

Procede para o fetch stage e adquire a próxima instrução do programa corrente no processo em execução

Se uma interrupção está pendente no processador:

Define o program counter para o endereço inicial de um programa de interrupt handler

Comuta do user mode para o kernel mode de modo que o código de processamento da interrupção possa incluir instruções privilegiadas

Modo de Comutação de Processos - 2

Os passos numa comutação de processos são:

Guardar o contexto do processador



Atualizar o PCB do processo correntemente no estado Running



Mover o PCB deste processo para a fila adequada



Seleccionar outro processo pra execução



Atualiza o PCB do processo selecionado



Atualizar as estruturas de gestão de memória



Restaurar o contexto do processador para o que existia ao tempo em que o processo selecionado foi trocado da última vez

Se o processo correntemente em execução vai ser passado para outro estado (Ready, Blocked, etc.), então o OS deve proceder a mudanças substanciais no seu ambiente de trabalho

Implementação de processos – 1

Alguns campos de uma tabela de processos

Process management	Memory management	File management
Registers	Pointer to text segment info	Root directory
Program counter	Pointer to data segment info	Working directory
Program status word	Pointer to stack segment info	File descriptors
Stack pointer		User ID
Process state		Group ID
Priority		
Scheduling parameters		
Process ID		
Parent process		
Process group		
Signals		
Time when process started		
CPU time used		
Children's CPU time		
Time of next alarm		

Implementação de processos – 2

Resumo das operações de baixo nível que o OS leva a cabo quando ocorre uma interrupção

1. Hardware stacks program counter, etc.
2. Hardware loads new program counter from interrupt vector.
3. Assembly language procedure saves registers.
4. Assembly language procedure sets up new stack.
5. C interrupt service runs (typically reads and buffers input).
6. Scheduler decides which process is to run next.
7. C procedure returns to the assembly code.
8. Assembly language procedure starts up new current process.

Modelando a Multiprogramação

Utilização da CPU em função do número de processos em memória

