

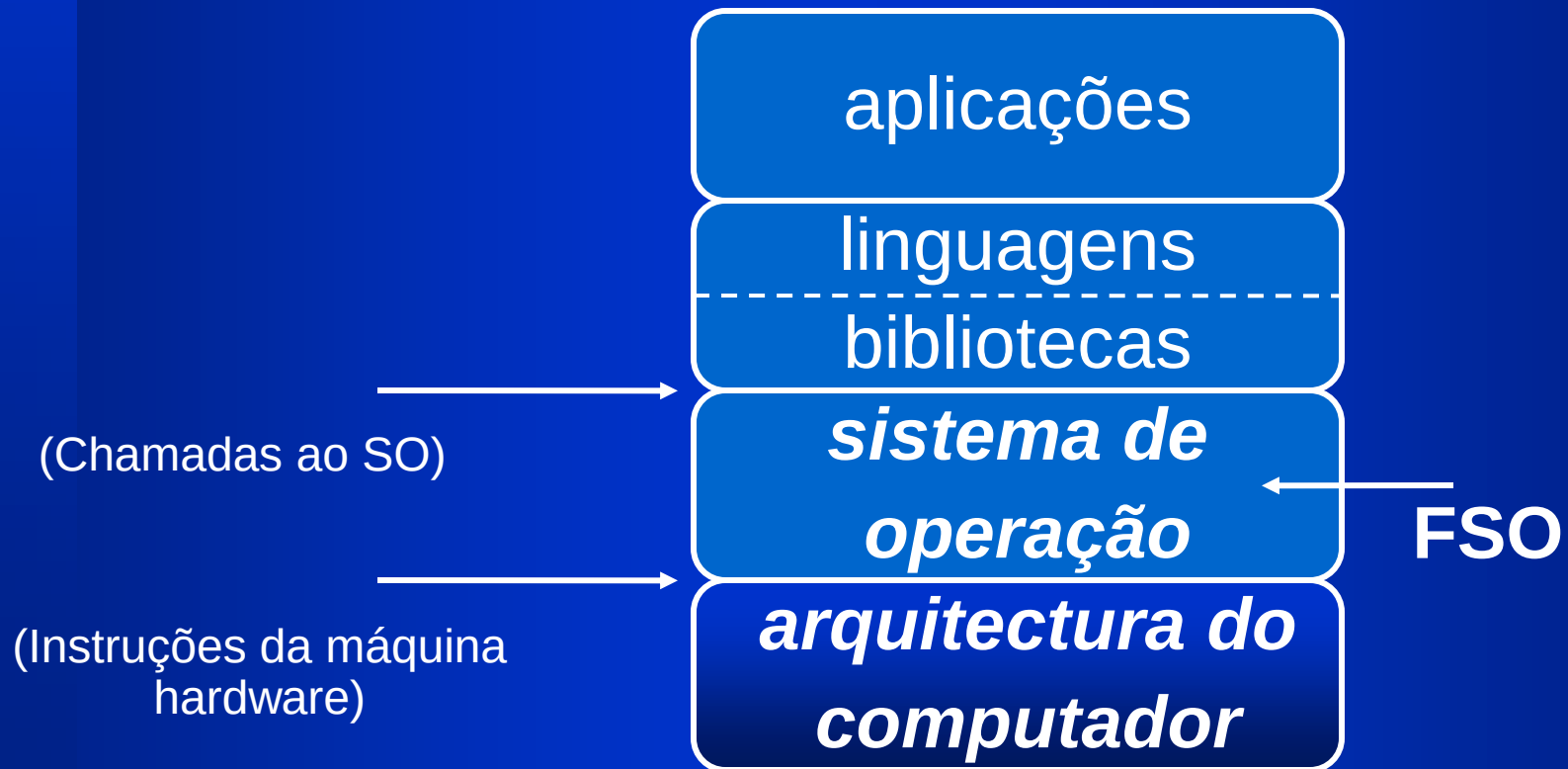
Fundamentos de Sistemas de Operação (FSO)

Lic. Eng. Informática 2012/2013

Responsável da disciplina: José A. Cardoso e Cunha

*Área Científica de Arquitectura de Sistemas Computacionais,
Dep. Informática – FCT/UNL*

Níveis de Arquitectura de um Sistema Computacional



FSO é parte da área científica de *Arquitectura de Sistemas Computacionais*

1o ciclo – Licenciatura em Eng. Informática

1o ano: **ISRC e AC**

2o ano: **FSO e Redes de Computadores**

3o ano: opcionais:

Codificação da Informação; Sistemas Distribuídos

2o ciclo – Mestrado em Eng. Informática

Algoritmos e Sistemas Distribuídos

Sistemas de Operação

Computação Paralela e Distribuída

Segurança de Sistemas e Redes de Computadores

Sistemas de Computação Móvel e Ubíqua

Sistemas e Tecnologias de Middleware

Redes de Computadores TCP/IP

Sistemas de Computação em Grid

Gestão de Centros de Dados

3o ciclo – Doutoramento em Eng. Informática

Arq. Computadores, Sist. Operação, Redes Computadores, Sistemas Distribuídos

Projectos CITI - Centro de Investigação em Informática e TI

Docentes

● Responsável da disciplina:

José Cardoso e Cunha jcc@fct.unl.pt

Professor Catedrático

Coordenador da Área de ASC do Dep. Informática

Agregado e Doutorado em Informática (UNL)

Licenciado em Eng. Informática (FCT-UNL)

Licenciado em Eng. Electrotécnica (IST-UTL) (Ramo Telecom. e Electrónica)

● Docentes das práticas:

Vítor Duarte vad@fct.unl.pt + Paulo Lopes poral@fct.unl.pt

Professores Auxiliares

Doutorados em Informática (UNL)

Mestres em Engenharia Informática (UNL)

Vítor Duarte: Licenciado em Eng. Informática (FCT-UNL)

Paulo Lopes: Licenciado em Eng. Electrotécnica (IST-UTL)

Programa de FSO

1. Objectivos e tipos de Sistemas de Operação (SO)
2. Conceitos de Processo e Fluxos de Execução Concorrentes
3. Programação Concorrente
4. Gestão dos Processadores e da Memória
5. Sistemas de Ficheiros e Gestão dos Dispositivos de I/O
6. Máquinas Virtuais e Gestão Global de Recursos
7. Interface de Programação a nível das Chamadas ao SO

Prática Laboratorial:

- Programação a nível das chamadas ao SO (Linux) envolvendo:
 - interacção com o utilizador (ao nível do *Shell*)
 - programação de processos concorrentes
 - comunicação e sincronização entre processos: memória partilhada, mensagens e sinais
 - acesso ao sistema de ficheiros

Funcionamento da disciplina

- **Aulas teóricas:** 2 turnos, 2 aulas / semana
 - Hora de início das aulas: à hora+10min.
 - Aulas de 3a-feira: intervalo de 10 min. ao fim de 50 min.
 - **Horário de dúvidas:**
 - José Cardoso e Cunha (ver na página)
- **Aulas práticas:** 7 turnos
 - Início na semana de 17 de Setembro
 - **Turnos:**
 - Vítor Duarte: P2, P3, P5, P6
 - Paulo Lopes: P4, P7, P8

Teóricas e Práticas: ?

- **Teóricas:**

- Compreender as ideias e discutir os conceitos

- **Práticas:**

- Fazer programas e experimentar, ilustrar os conceitos, compreender melhor...

Nas aulas práticas...

- Pratica-se...

- Fazendo: programas em C utilizando a interface de chamadas ao sistema Linux
- Utilizam-se os conceitos e técnicas de programação que são apresentados na teórica

Os trabalhos práticos incidem sobre a aplicação da matéria apresentada na aulas teóricas

Conselhos gerais

- Sigam as aulas e a matéria desde o início
- Façam os trabalhos práticos
- Façam os testes de auto-avaliação
(na página da disciplina, de 15 em 15 dias)
- Esclareçam as dúvidas DURANTE e não no fim do semestre...

Apoio à cadeira

- Informação on-line a partir de:
<http://asc.di.fct.unl.pt/~jcc/fso-12-13>
e na página do CLIP
- Livros e elementos de base
- Elementos complementares

Elementos de Estudo

Notas das Aulas Teóricas: jcc, DI/FCT/UNL
(cobrem toda a matéria dada; acessíveis na página Web junto dos sumários das aulas)

Notas das Aulas Práticas:
(apoiam os trabalhos práticos; acessíveis na página Web)

LIVROS recomendados:

Operating System Concepts - Essentials, A. Silberschatz, P. B. Galvin, G. Gagne, John Wiley & Sons, 2010.

Unix System Programming, K. Haviland, D. Gray, B. Salama, Addison-Wesley, 2nd ed., 1999

Sistemas Operativos, J.Alves Marques et al, FCA-Ed.Inf, 2012

Avaliação

- **Práticas laboratoriais:**

- realização de 2 trabalhos, grupos 2 alunos
- nota **NP média ponderada de notas dos trabalhos práticos entregues nos prazos**

- **Frequência para obter Nota Final na disciplina: NP $\geq$ 9.5**

- **Frequência** obtida nos anos anteriores:

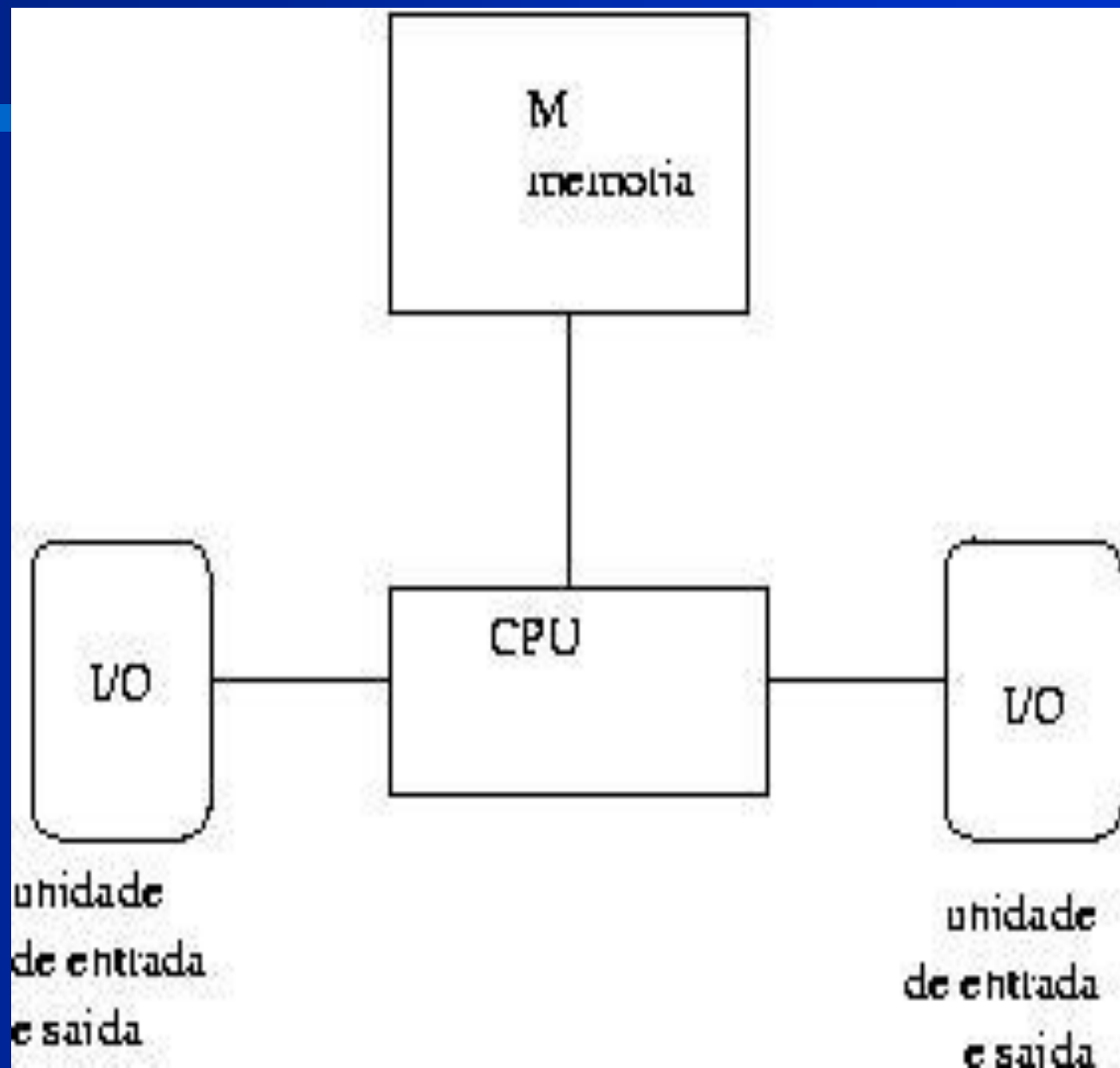
- mantém-se para este ano (nota final dada pelos testes)
- se quiserem, podem acompanhar os trabalhos – falar com docentes das práticas

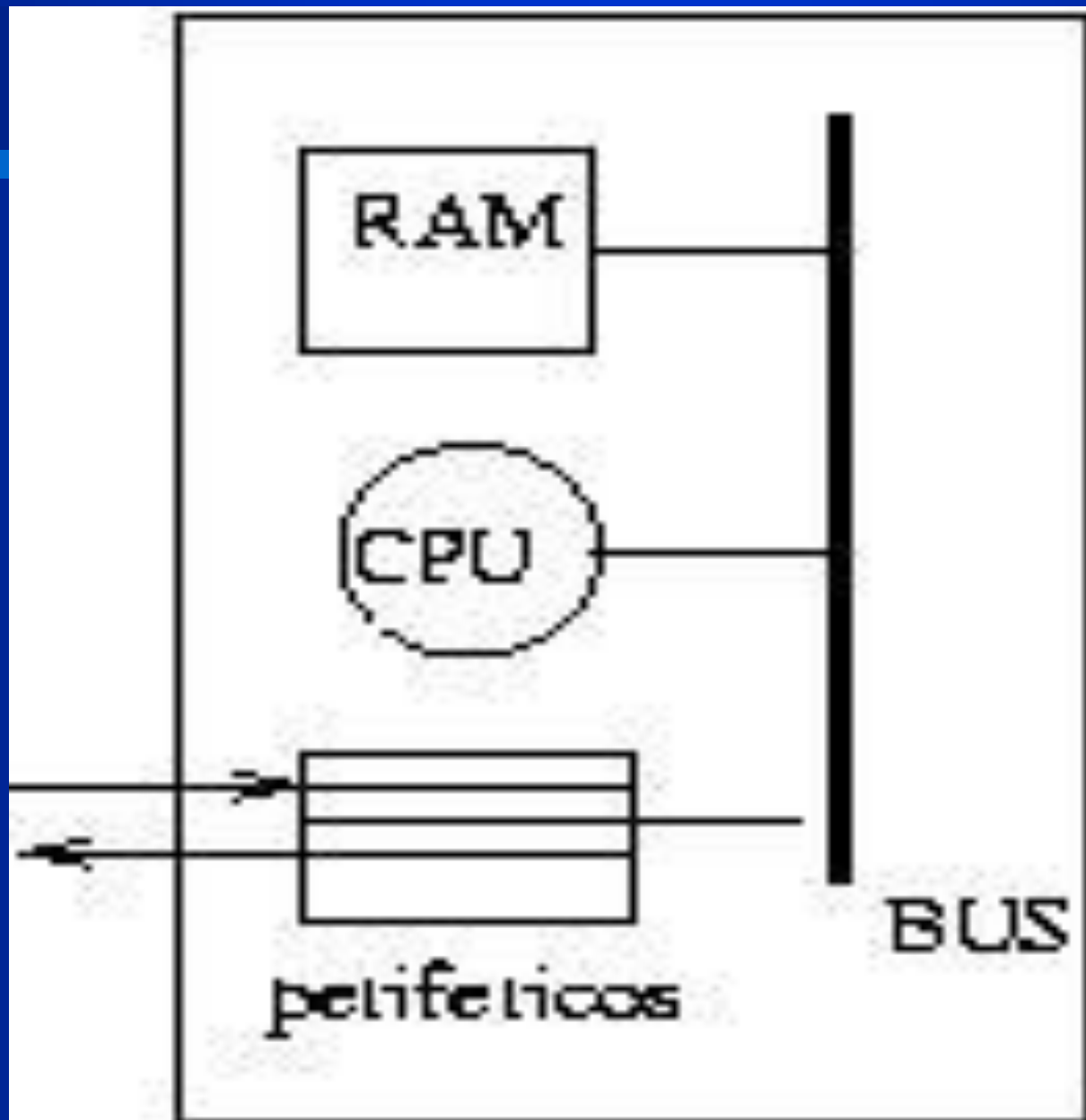
Avaliação

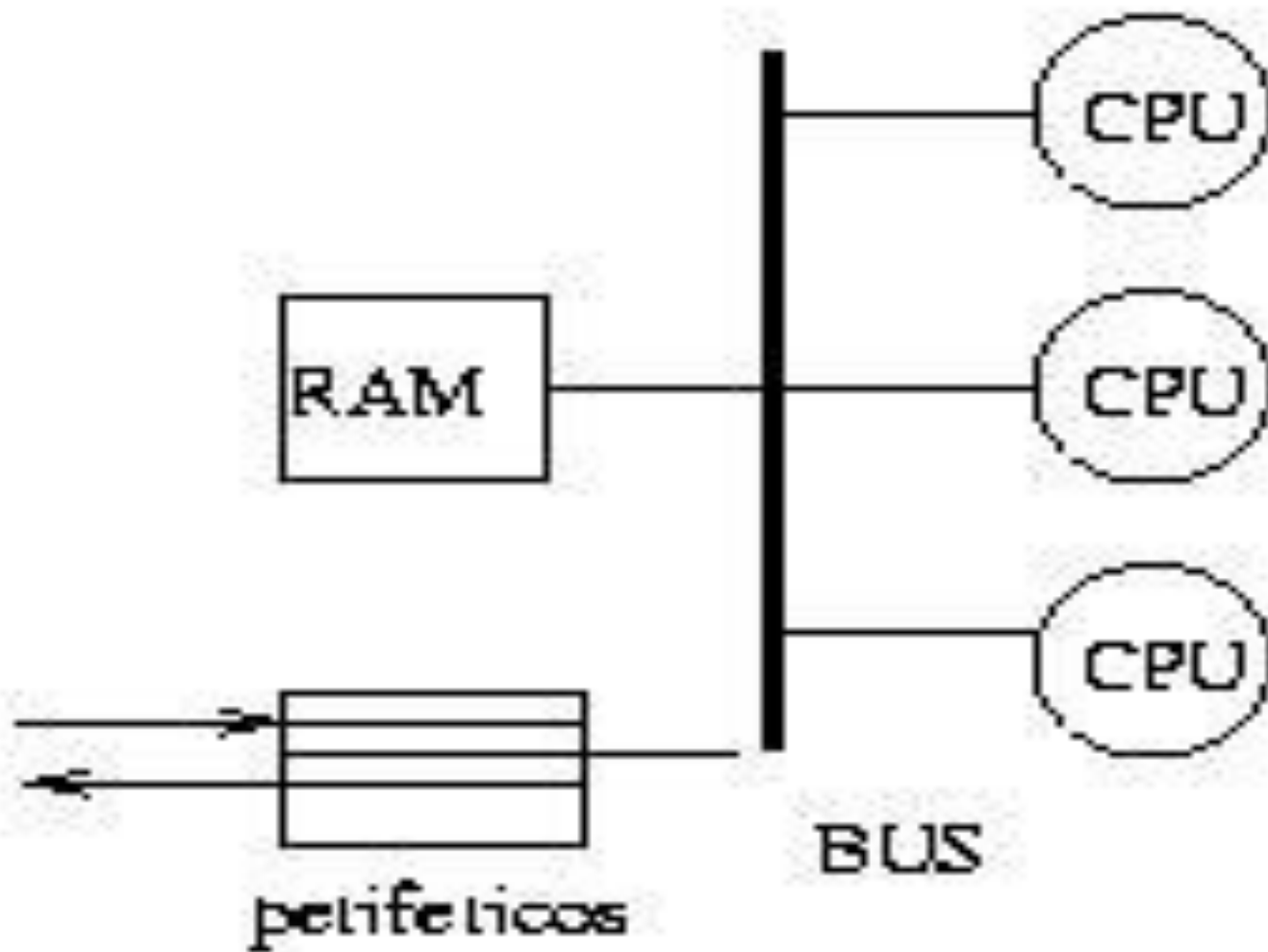
- **Nota Final = $0.70*NT + 0.30*NP$**
- **NT: 2 Testes:**
 - Testes: **NT = média aritmética dos testes**
Se **tem Frequência e Nota Final ≥ 9.5** após os testes:
 - Nota Final
 - Pode fazer Melhoria Nota (inscrição RepAcadémica)
 - Se não passou pelos testes, faz o exame:
NP mantém-se
- **Datas dos testes: (previstas- aguarda CC)**
 - T1: semana 22 Outubro 2012
 - T2: semana 3 Dezembro 2012

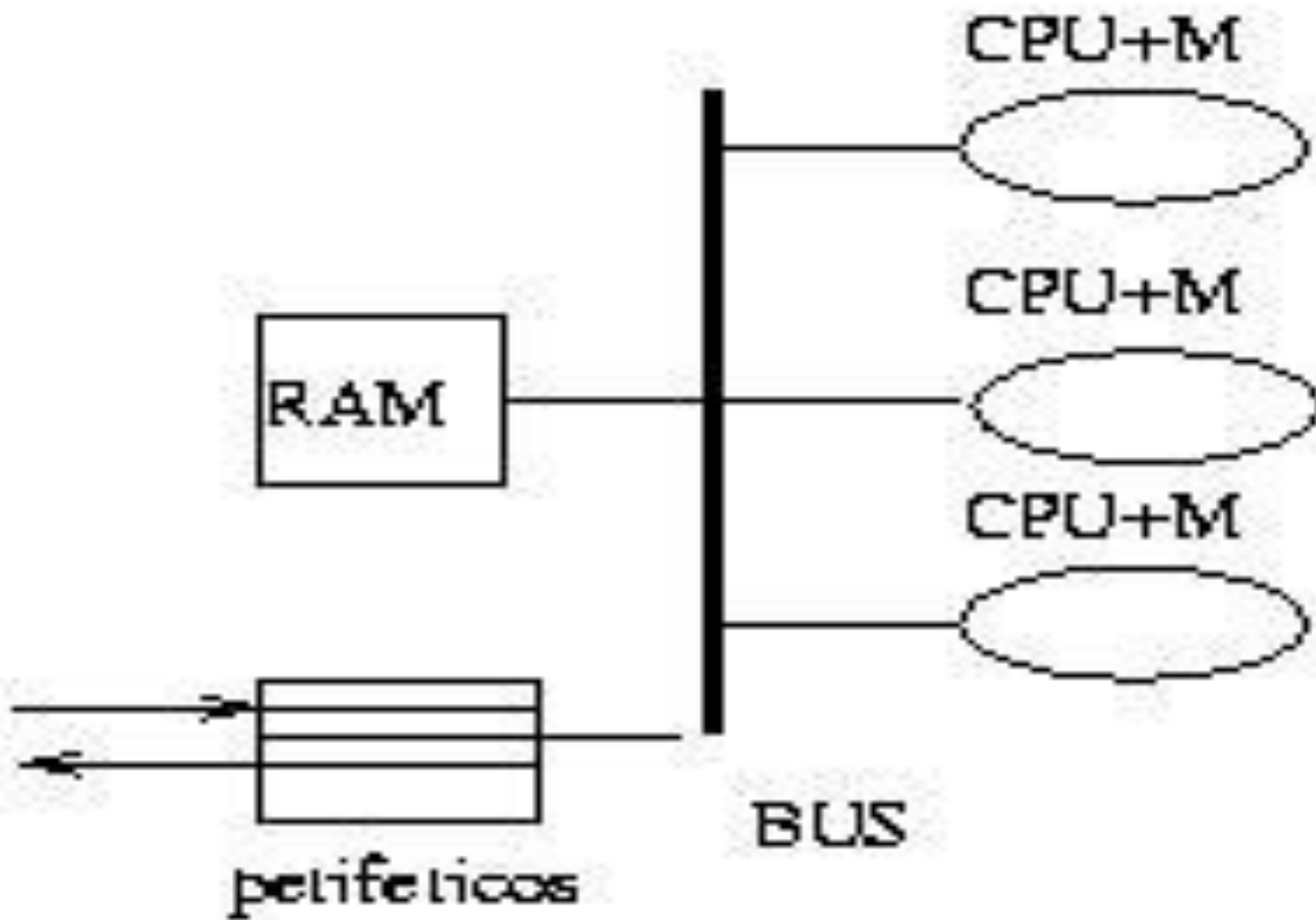
SO, numa série de figuras...

Arquitectura de Computadores...

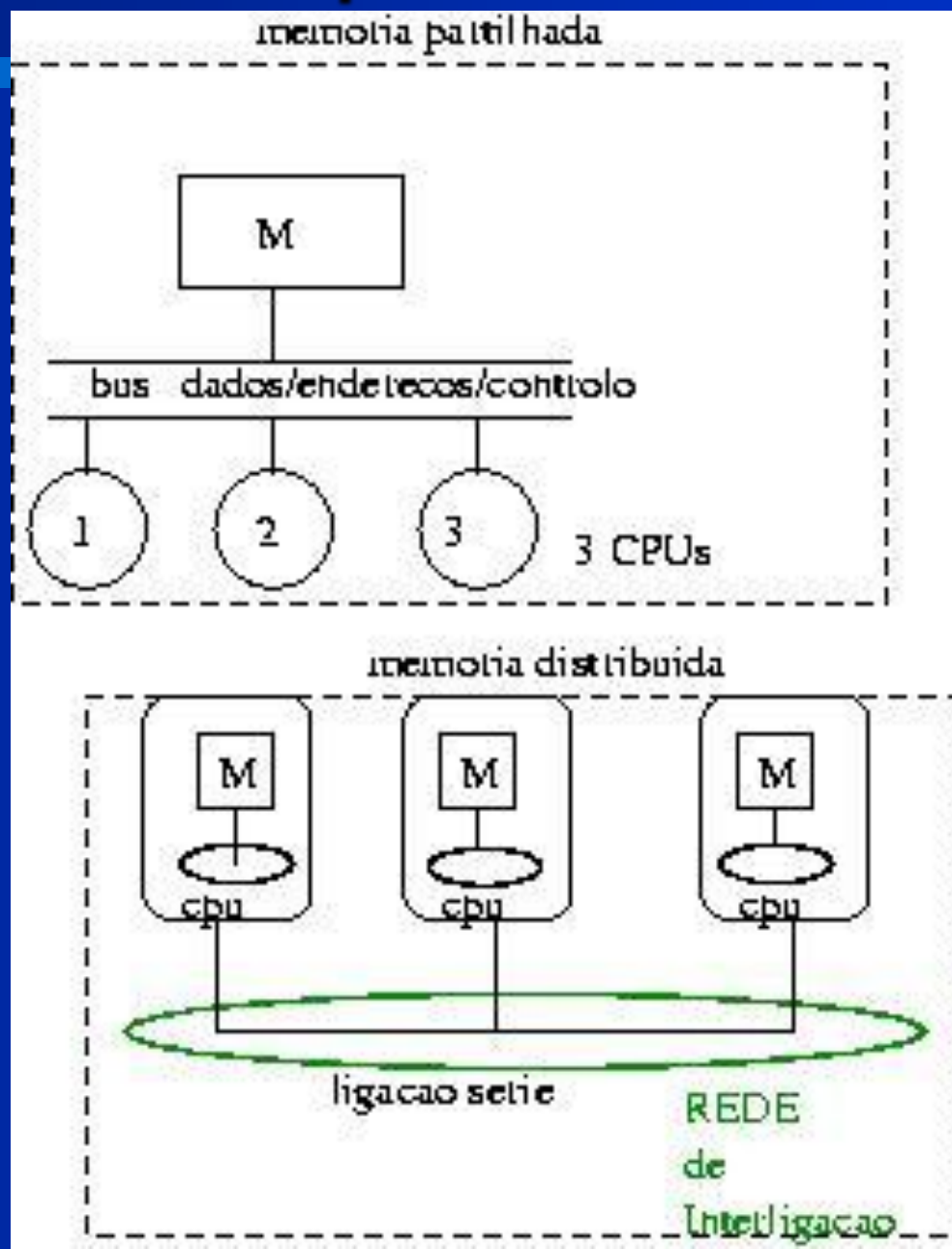


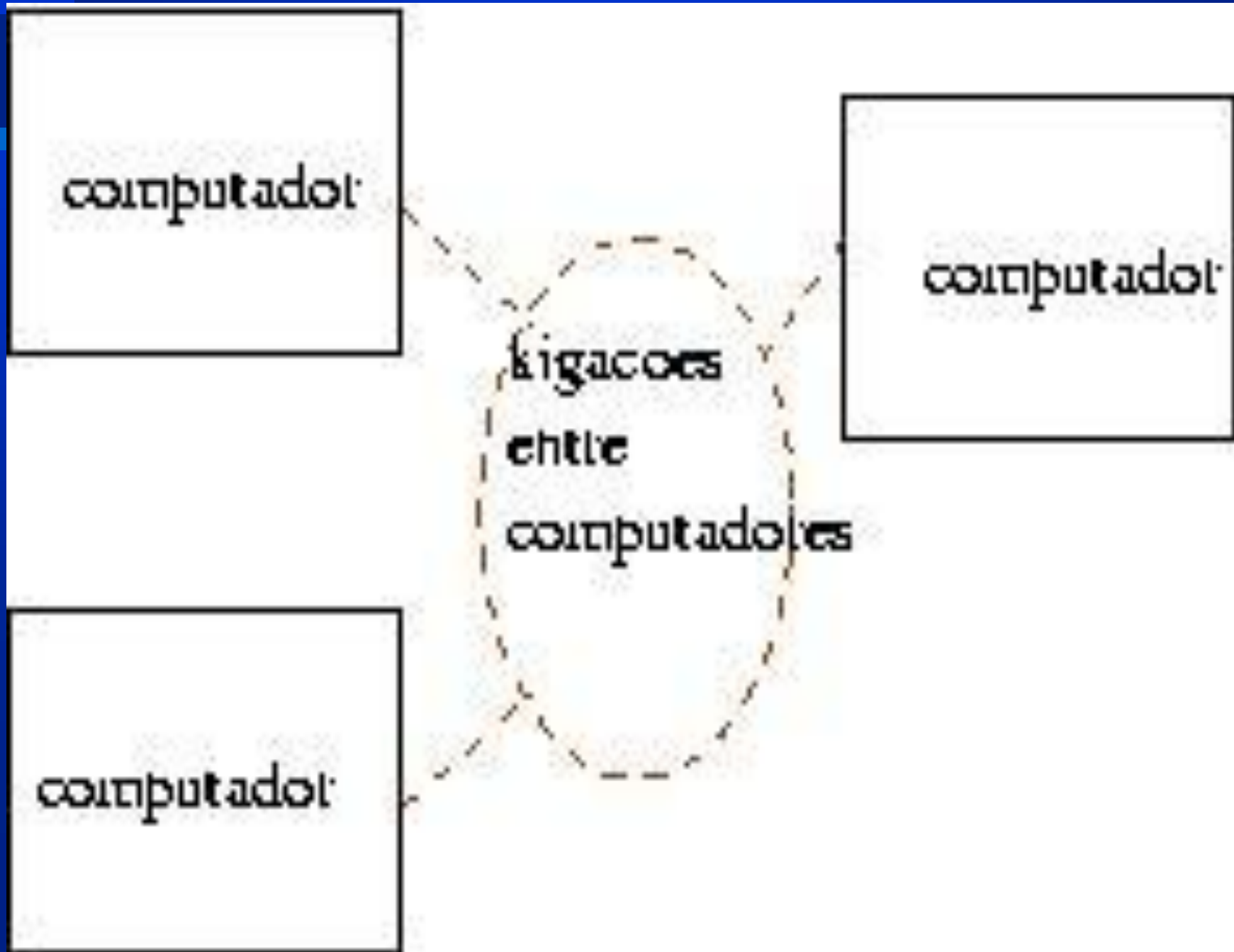




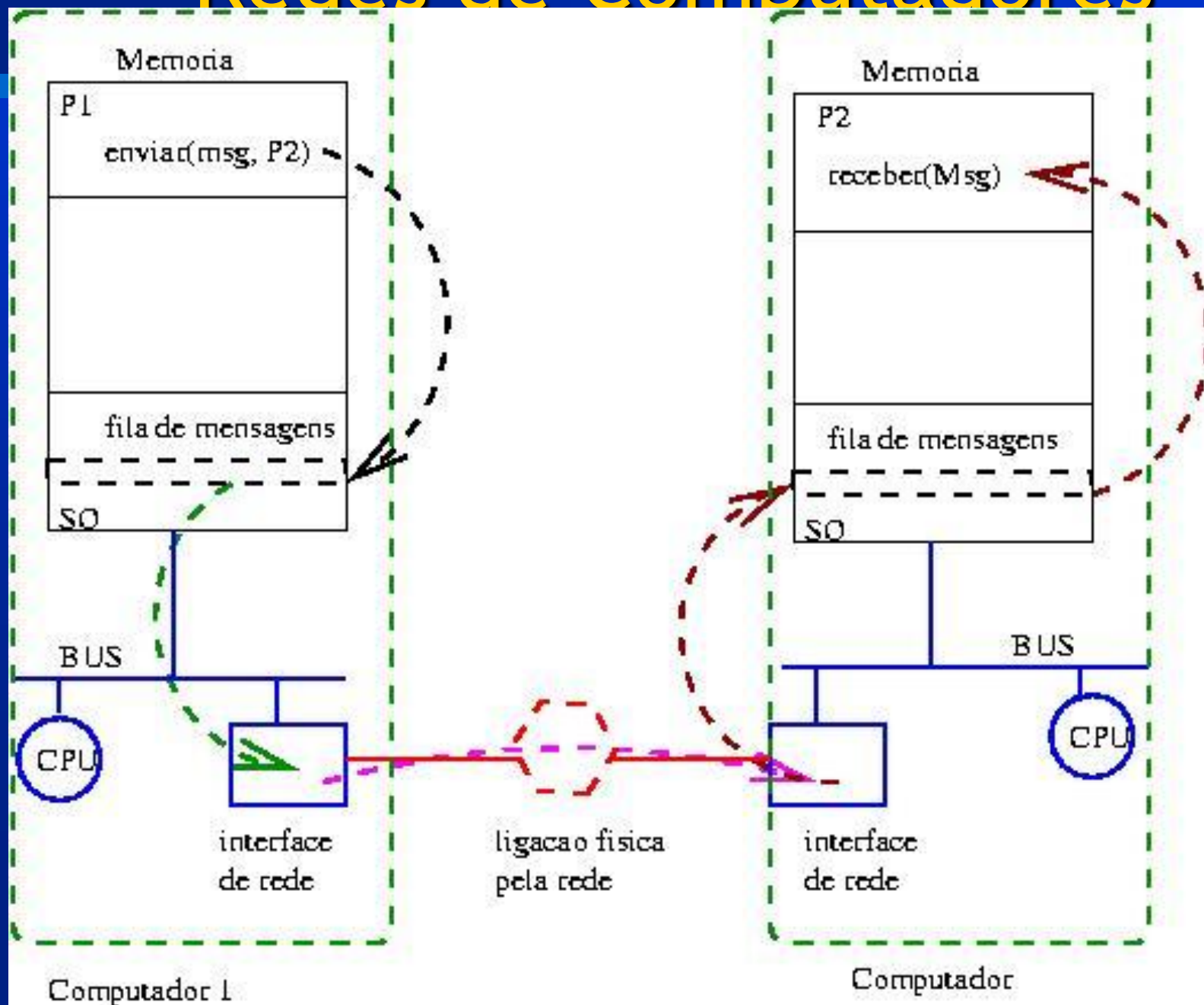


Multiprocessadores





Redes de Computadores



O Sistema de Operação visto como...

● Gestor de Recursos

- Gere e permite a partilha do *hardware* pelos processos
- Gere a execução dos programas e as suas interacções
- Garante a protecção e a consistência no acesso aos recursos (processamento, memória, comunicação)

● Máquina Virtual

- Suporta uma interface de programação
- Operações de mais "alto nível" que suportam o resto do sistema e que são chamadas pelos programas utilizadores

Principais grupos de operações

- **Controlo da execução** de programas
 - Iniciar, detectar erros, terminar, gerir a memória, etc.

Os *processos*
- Operações de entrada/saída de dados (ou I/O) e arquivo
 - Comunicar com os periféricos...
 - O acesso à informação em disco

Os *canais de comunicação e o sistema de ficheiros*
- **Concorrência** entre processos
 - Os problemas de escalonamento e de sincronização
 - Multiprogramação: múltiplas aplicações / utilizadores
- **Comunicação** entre processos concorrentes
 - A transmissão de informação (mensagens)
 - O acesso à informação partilhada

SO

Application
Programs

Application
Programs

User Mode

Kernel Mode

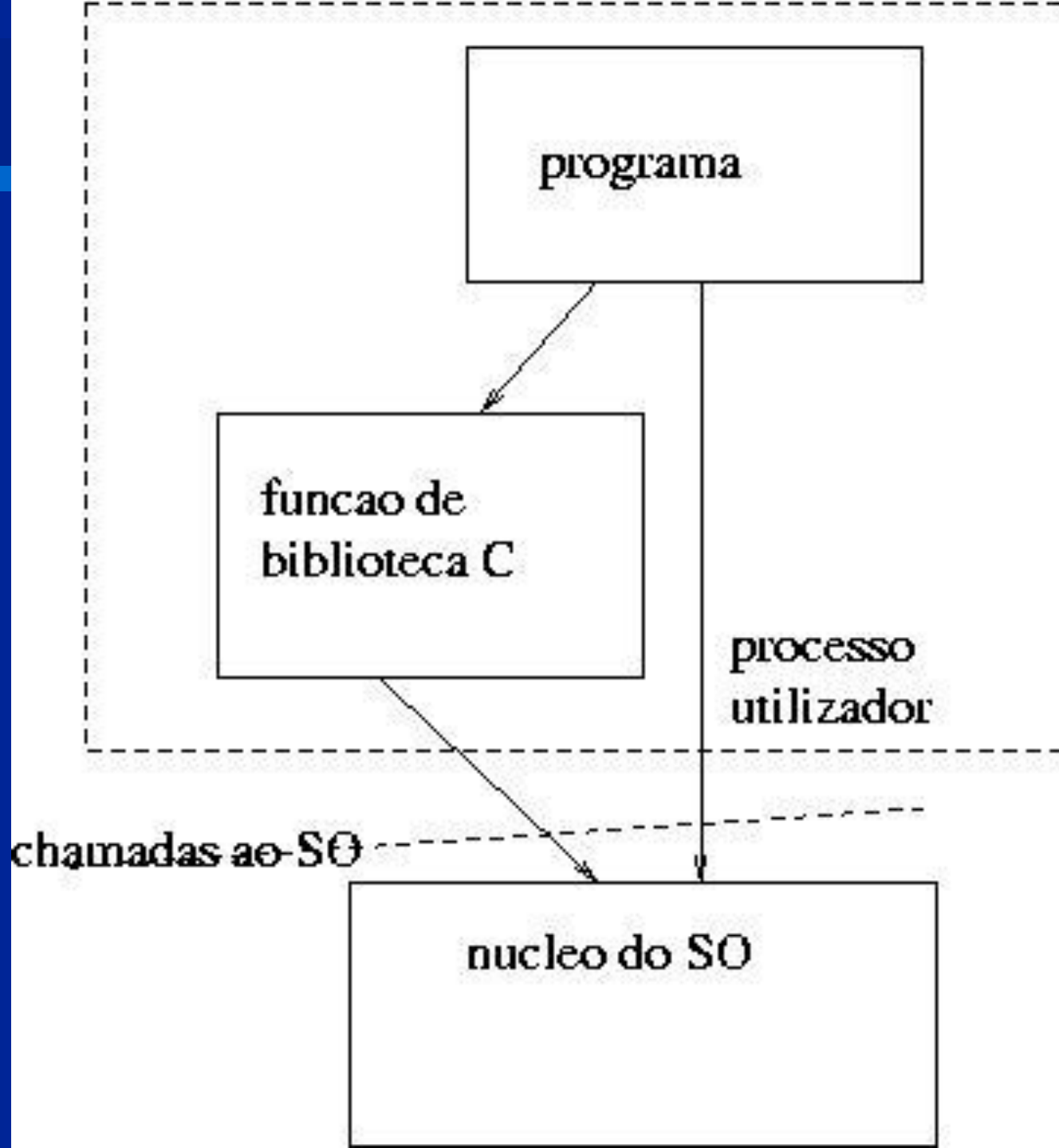


Hardware

Núcleo do SO

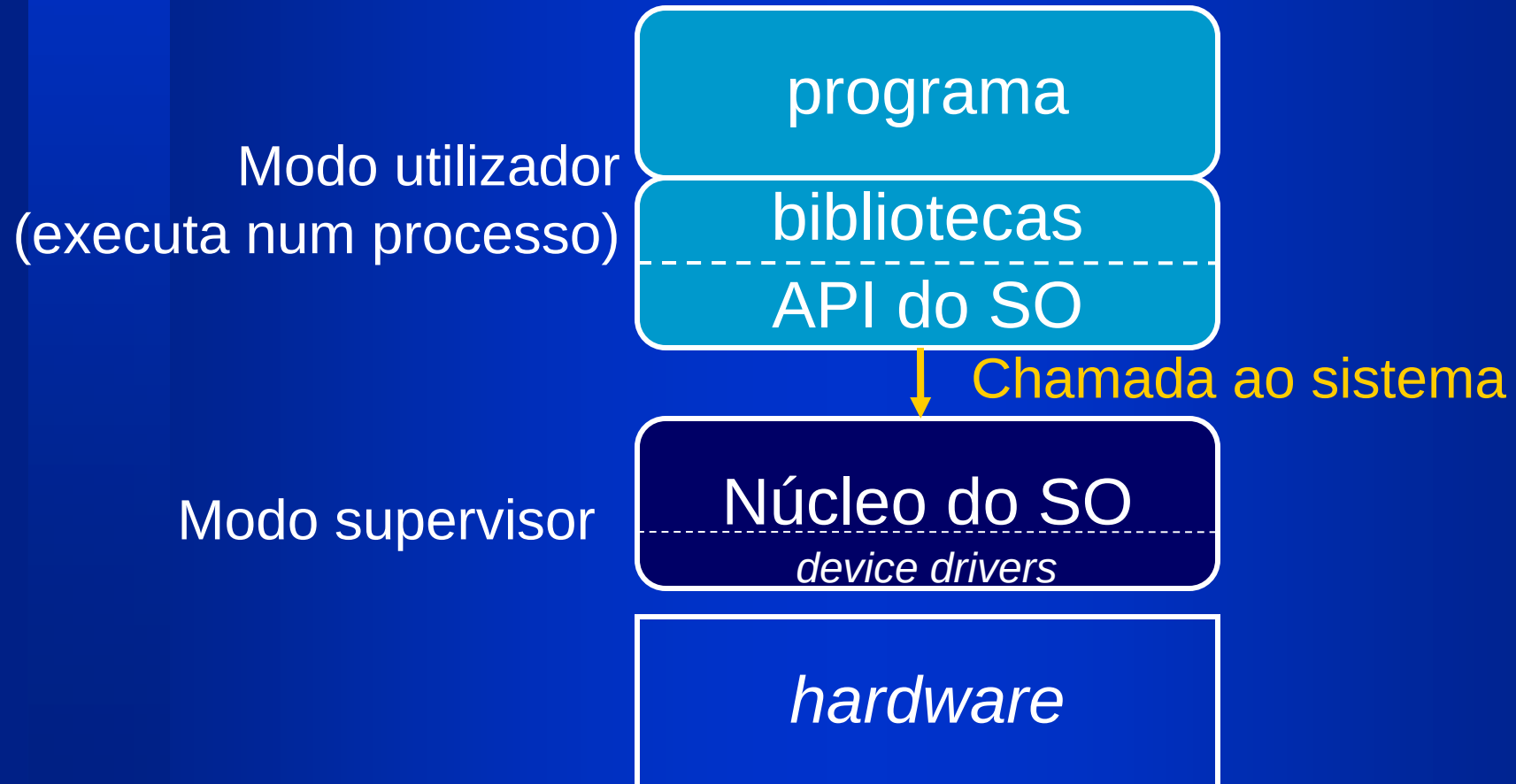
- O **Núcleo** suporta mecanismos básicos para controlo da execução dos programas e comunicação com o exterior (outros processos e dispositivos periféricos),
 - programas que se executam em modo privilegiado
 - capaz de controlar o *hardware* (CPU, M, periféricos)
- **Sobre o Núcleo**, os programas (em modo utilizador) fazem pedidos ao SO, por invocação de
 - chamadas ao sistema (*System Calls*)





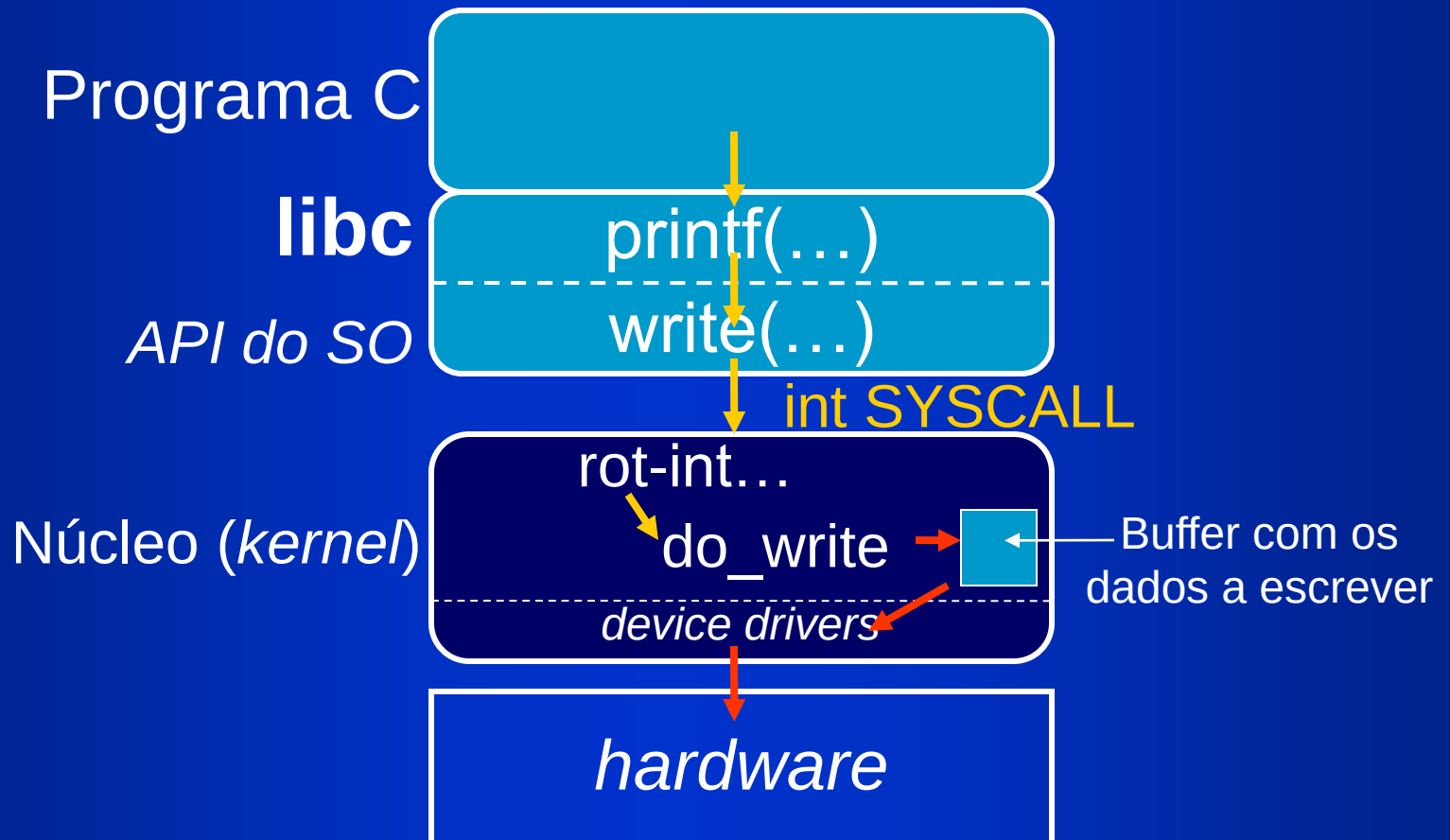
Níveis de software

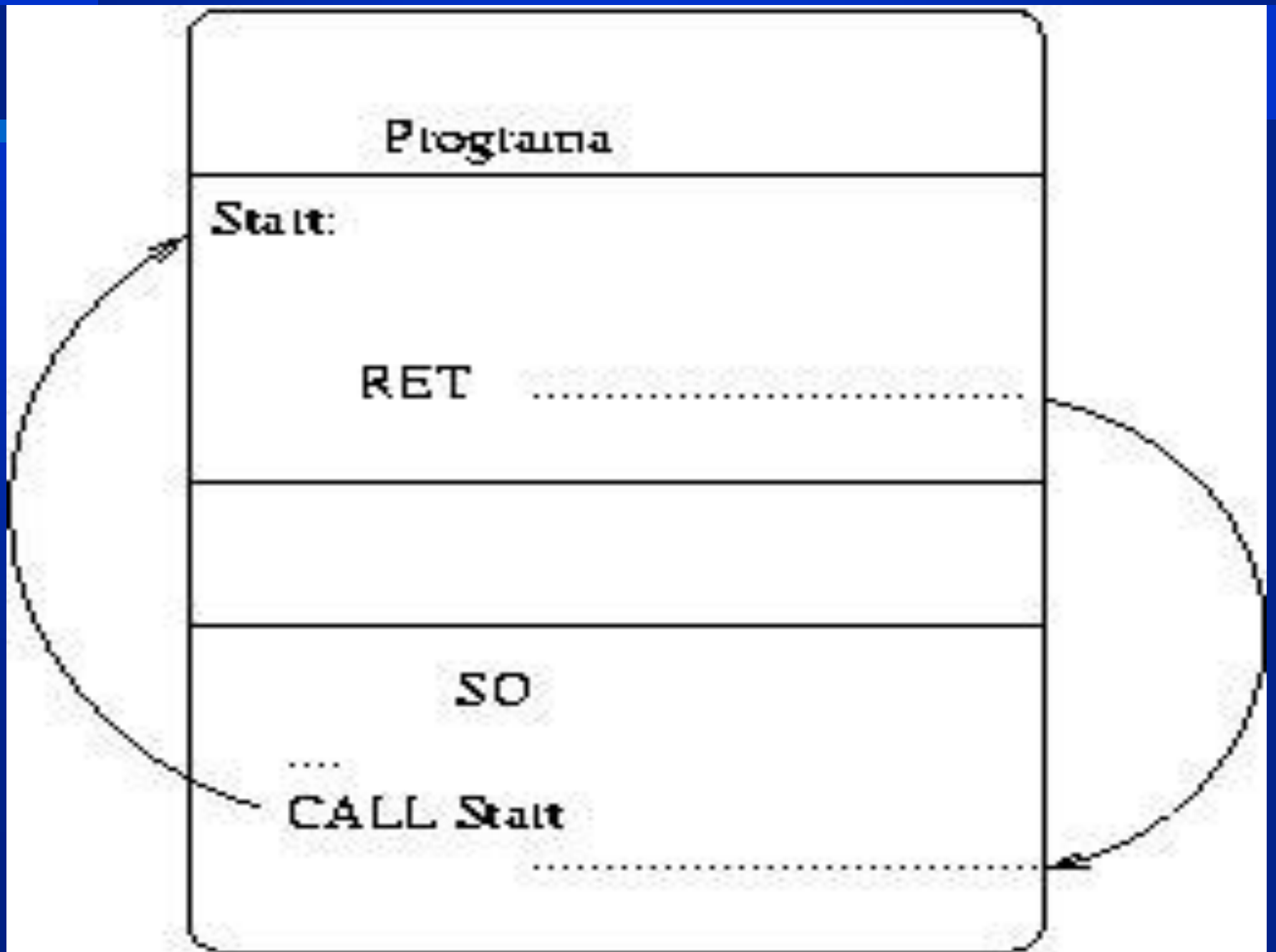
- Visão de uma aplicação



Chamadas ao sistema (I/O)

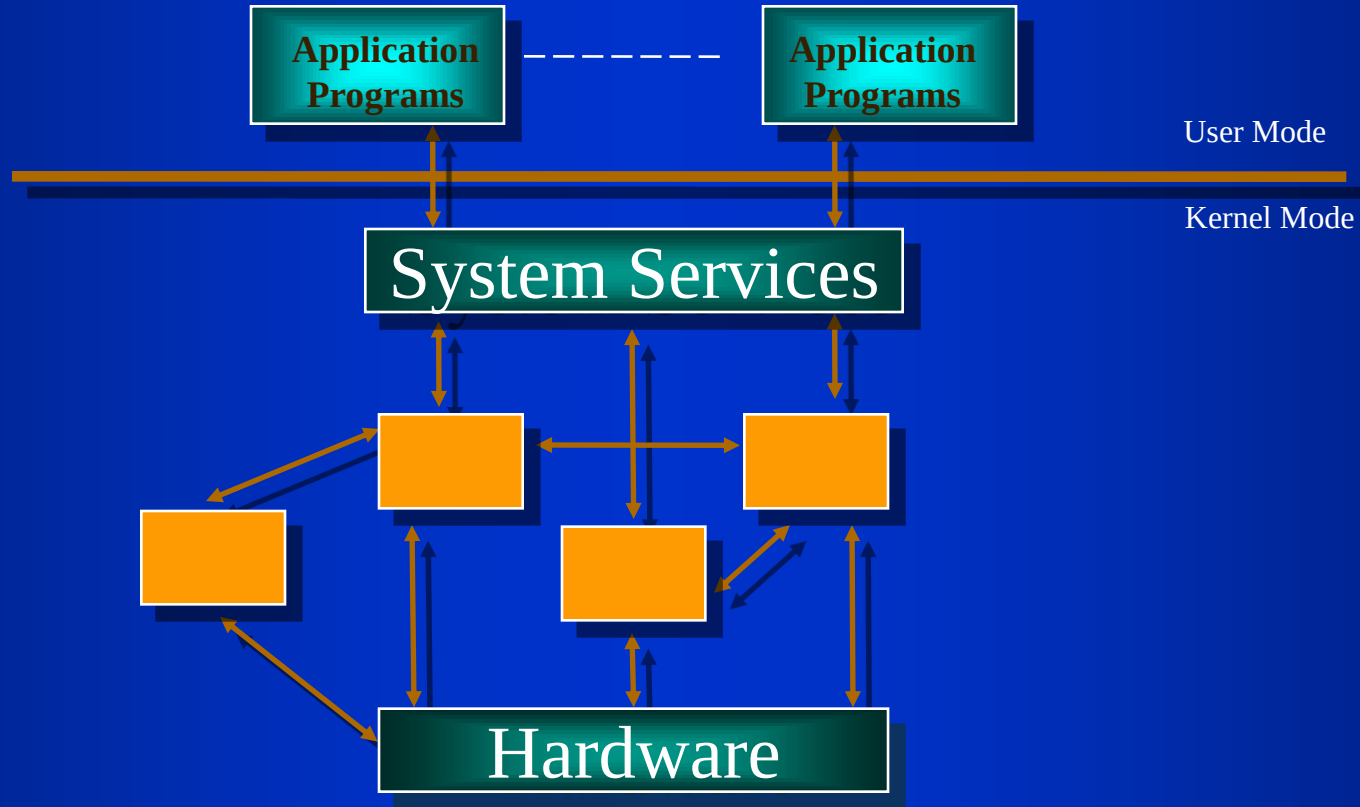
- Exemplo de saída de dados



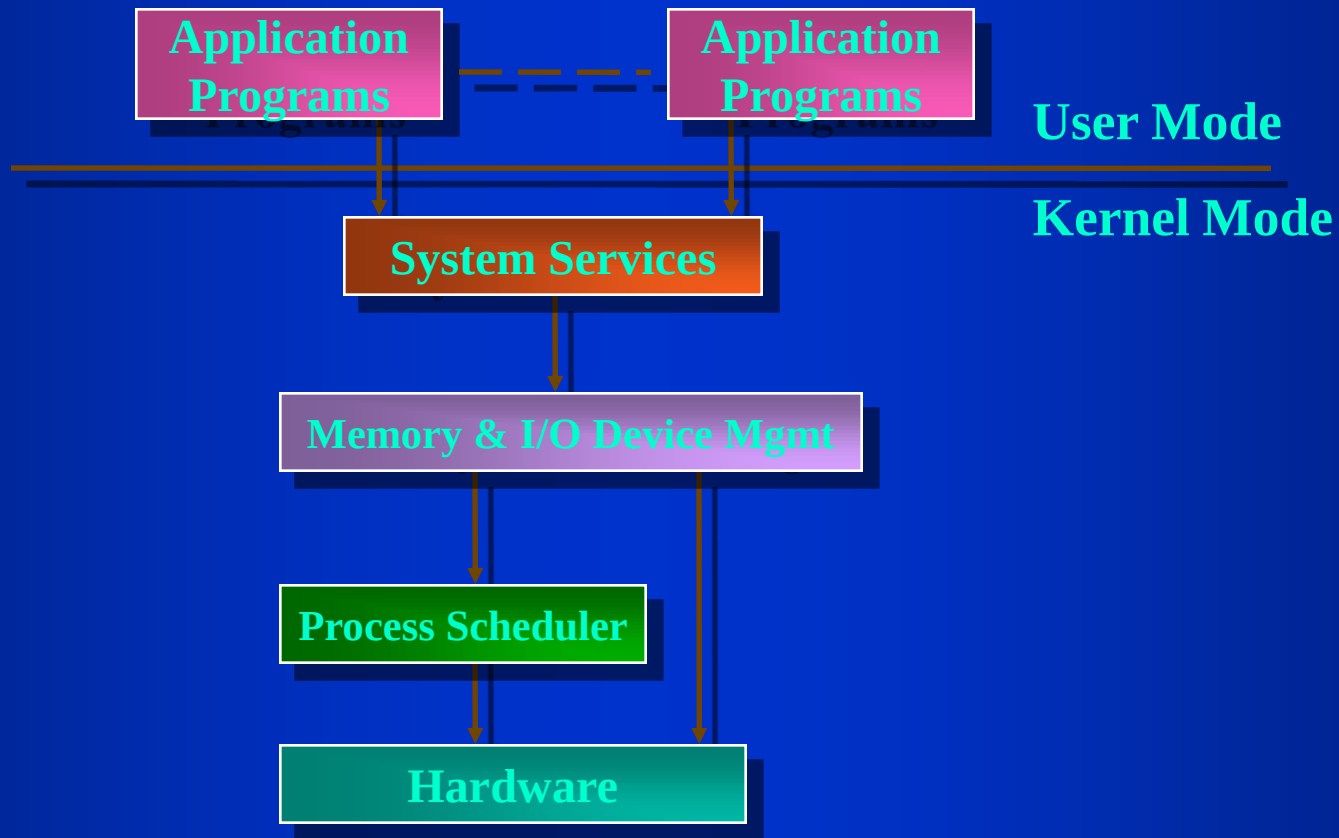


SO: organização interna...

SO



SO: camadas internas



Cada camada acede a uma interface de mais baixo nível

Ex : UNIX

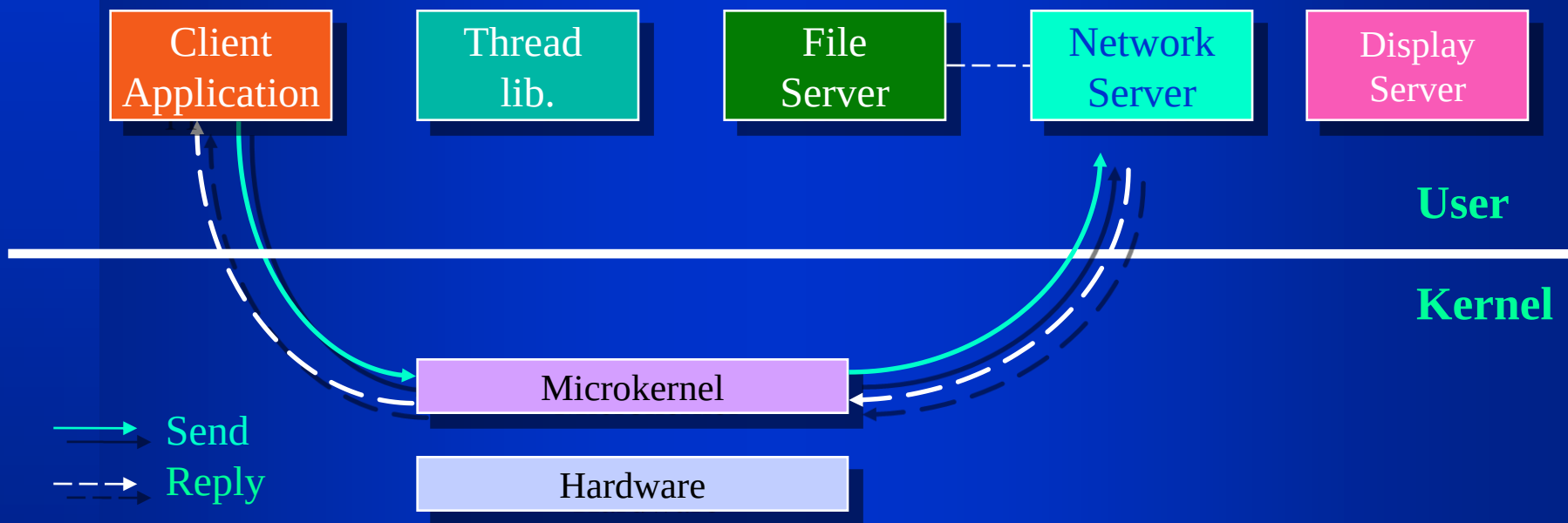
Núcleo do SO + Serviços

- **No Núcleo:** só as funcionalidades básicas essenciais
- As outras: suportadas sob a forma de **Serviços**, executados por processos (em modo utilizador).

Tendência para SO mais flexíveis

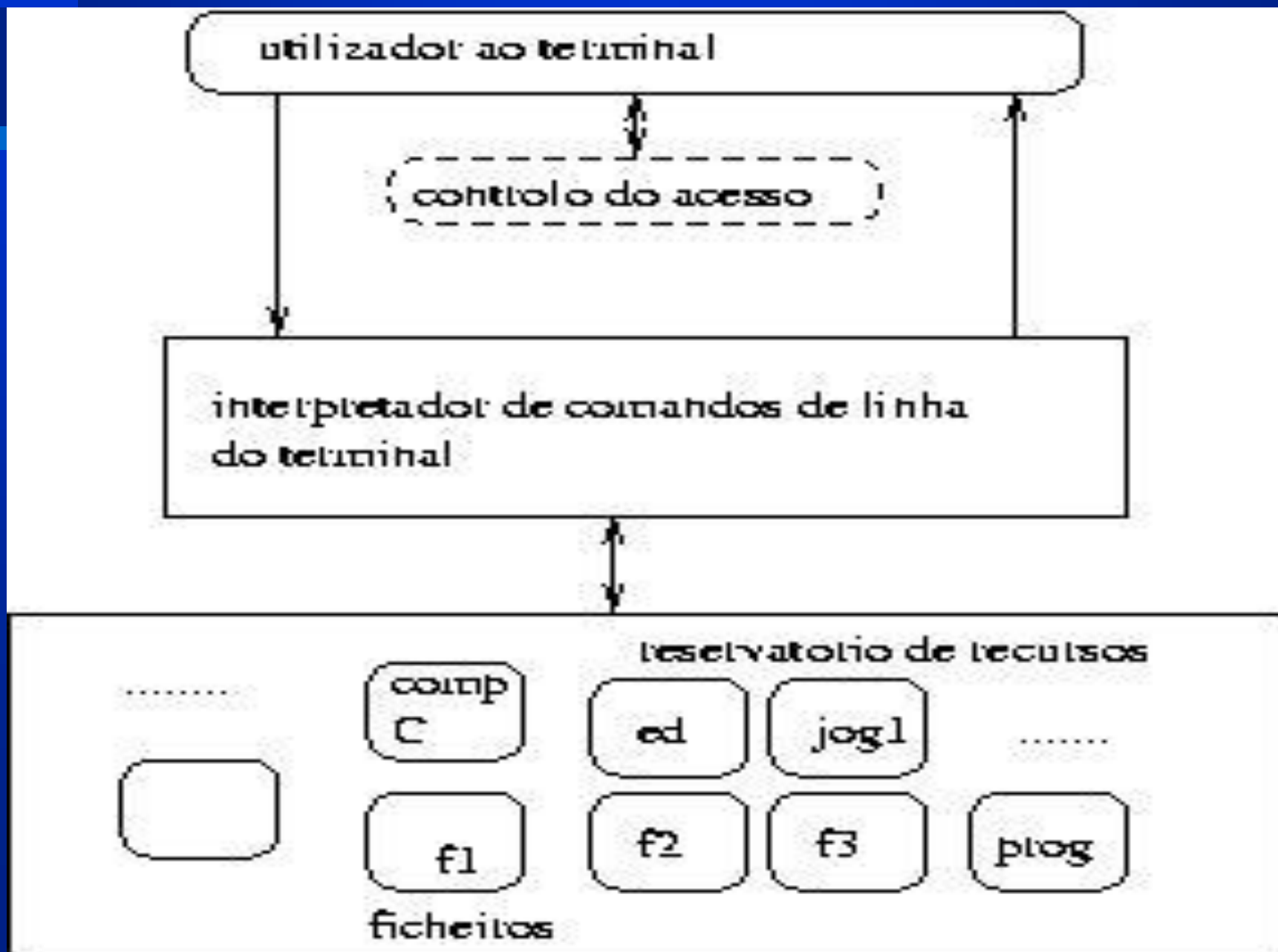


SO = núcleo mínimo + serviços

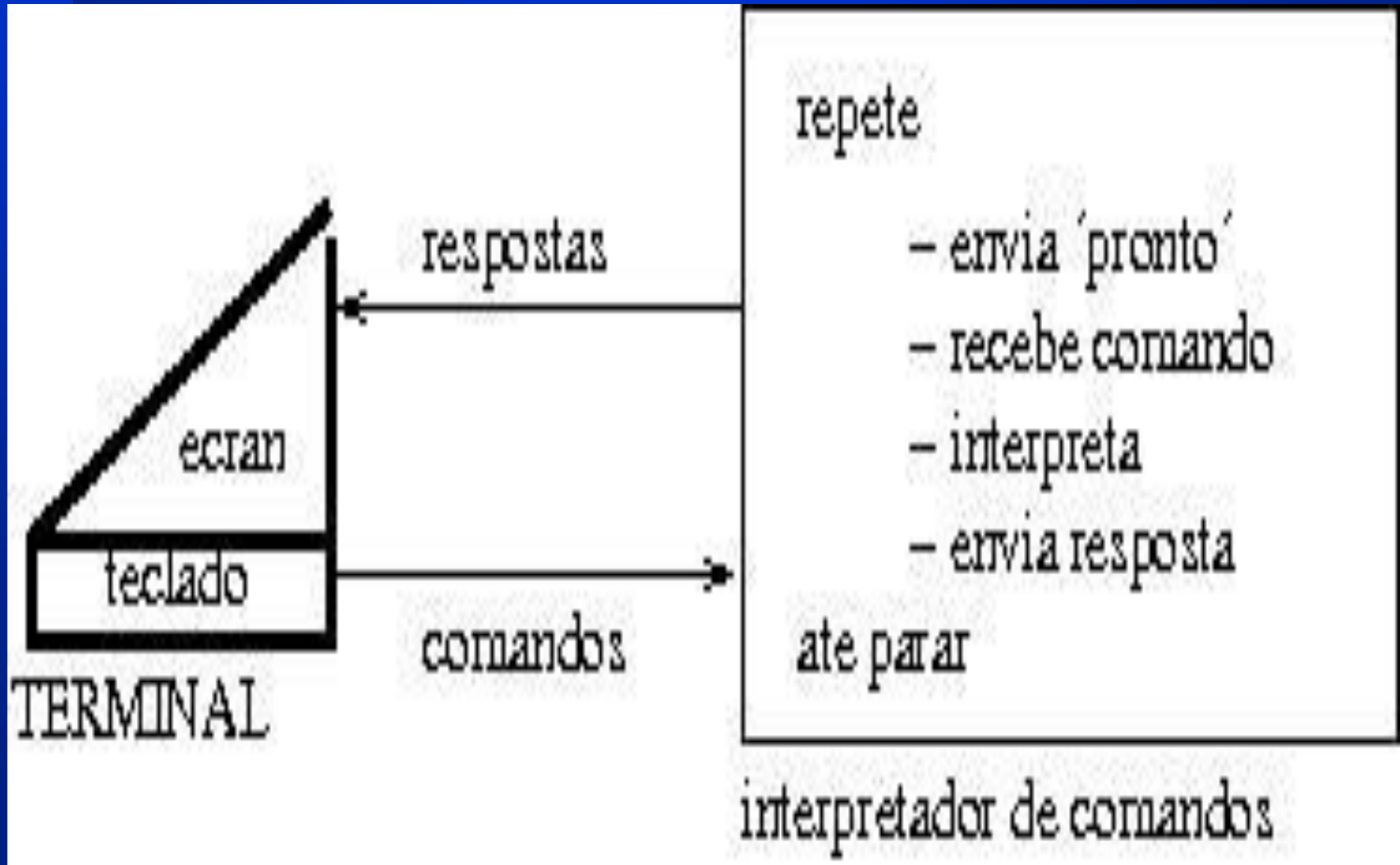


- ❑ Reduzido núcleo oferece primitivas básicas (processos, memória, I/O, comunicação)
- ❑ Serviços tradicionais tornam-se subsistemas
- ❑ OS = Microkernel + User Subsystems

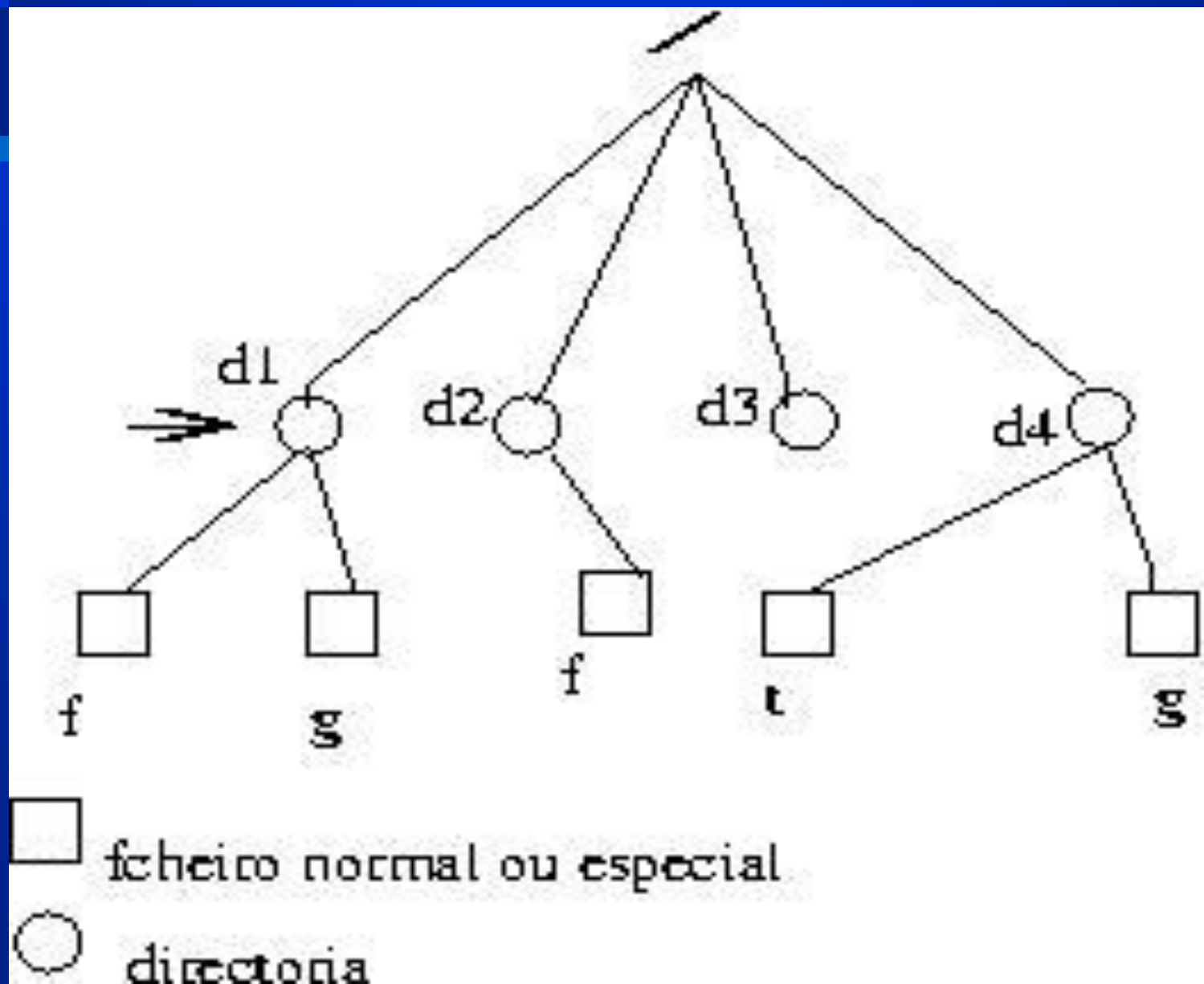
Perspectiva do Utilizador...

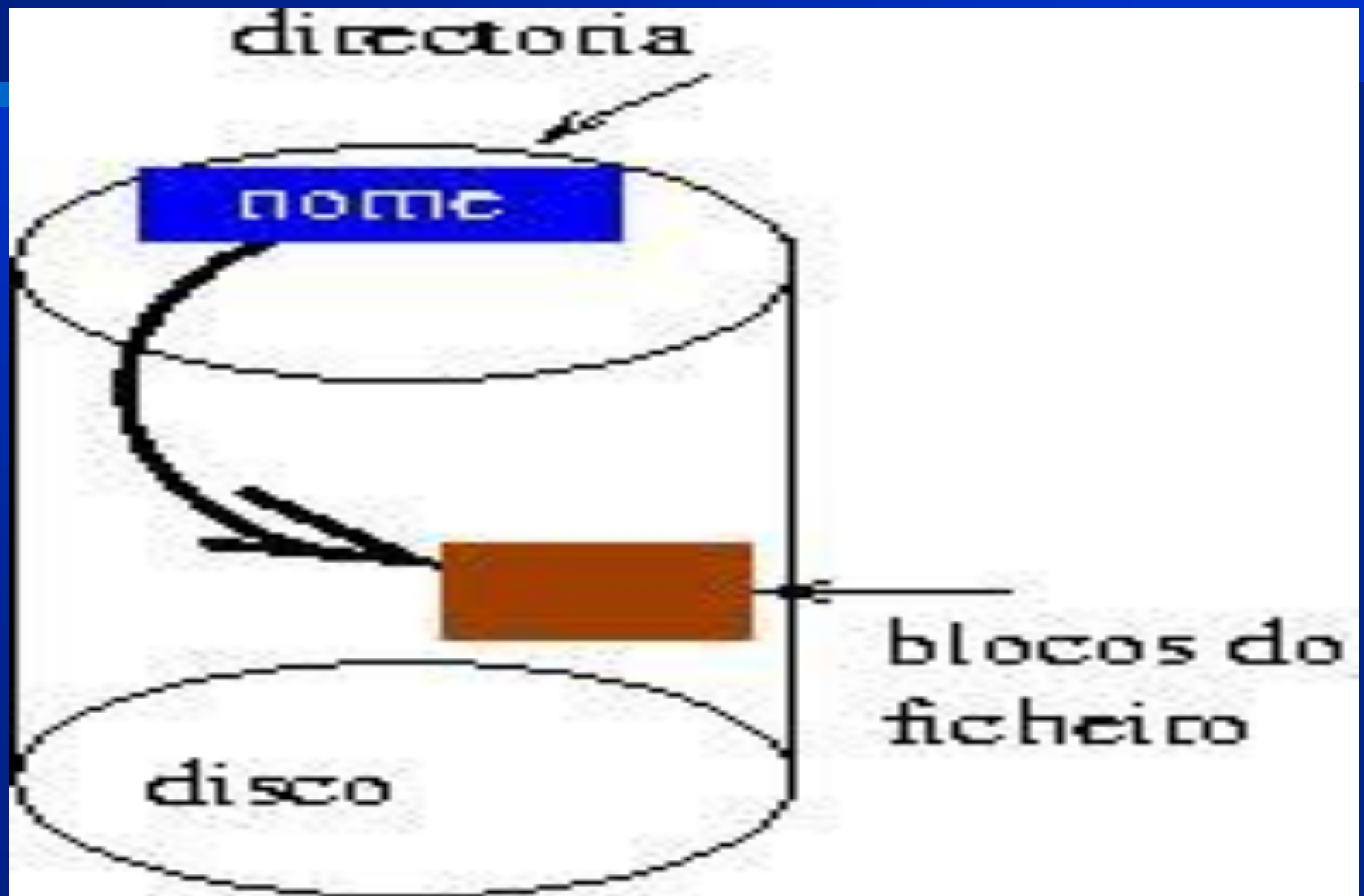


Interpretador de comandos



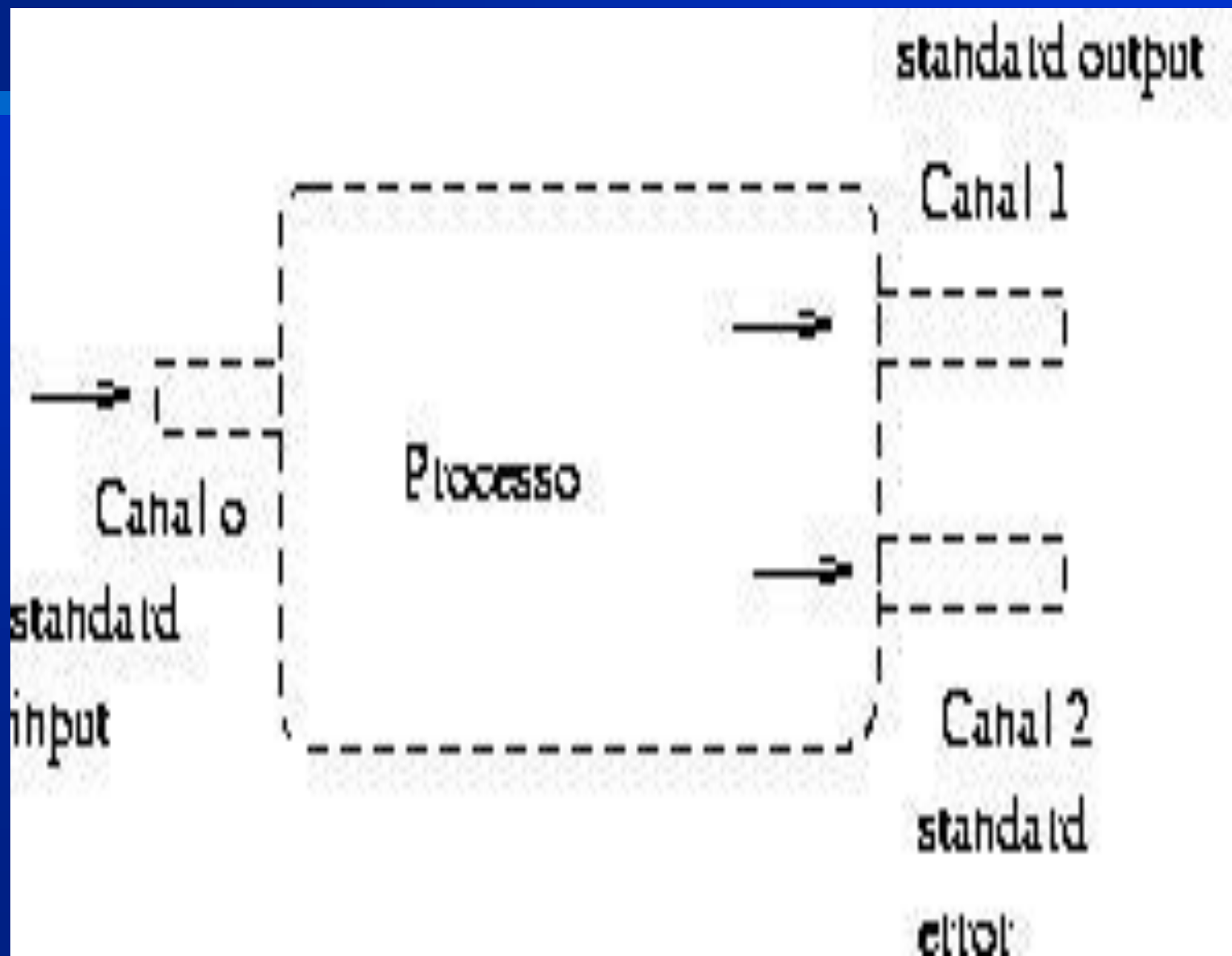
O Sistema de Ficheiros...



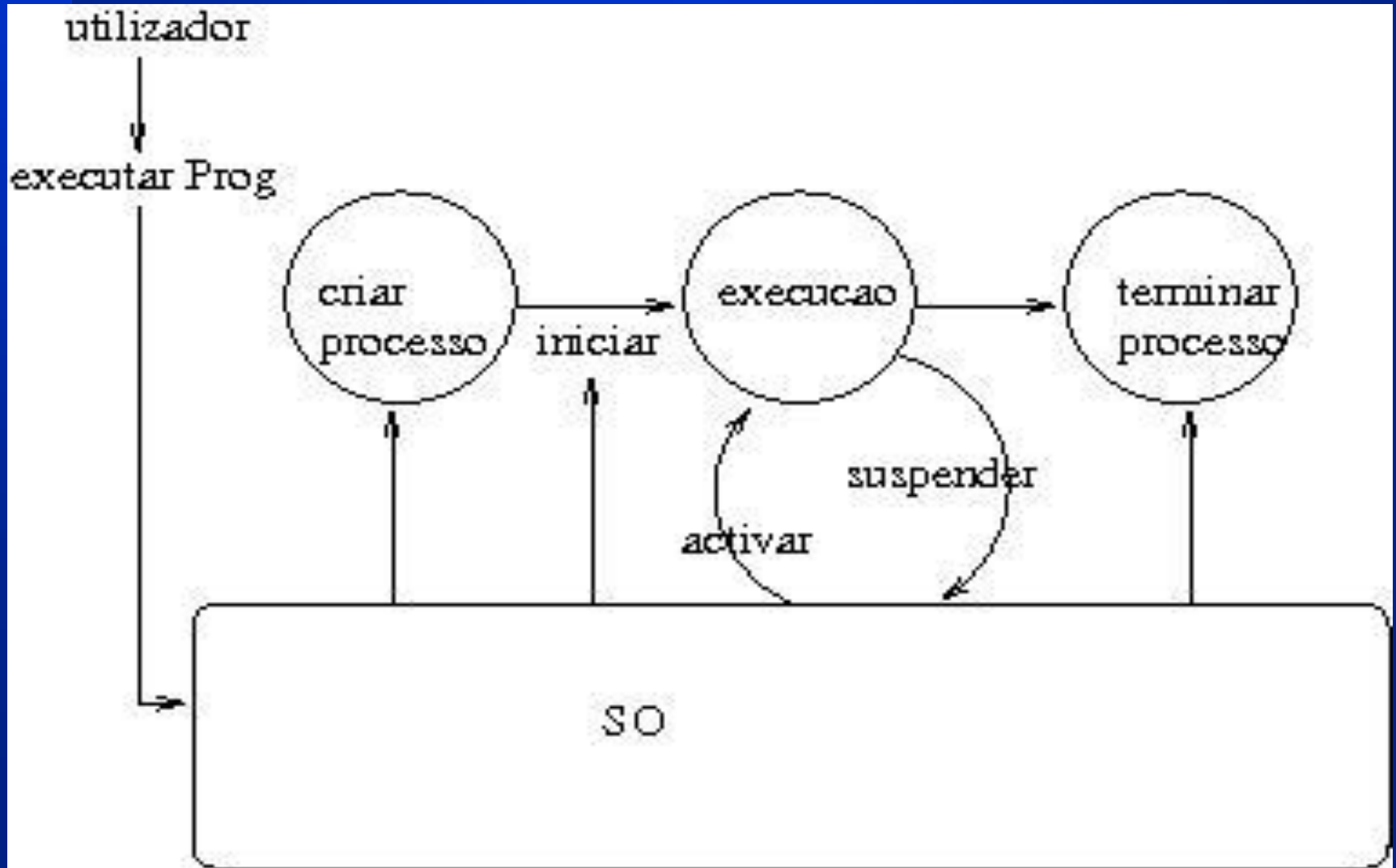


Controlo dos programas...

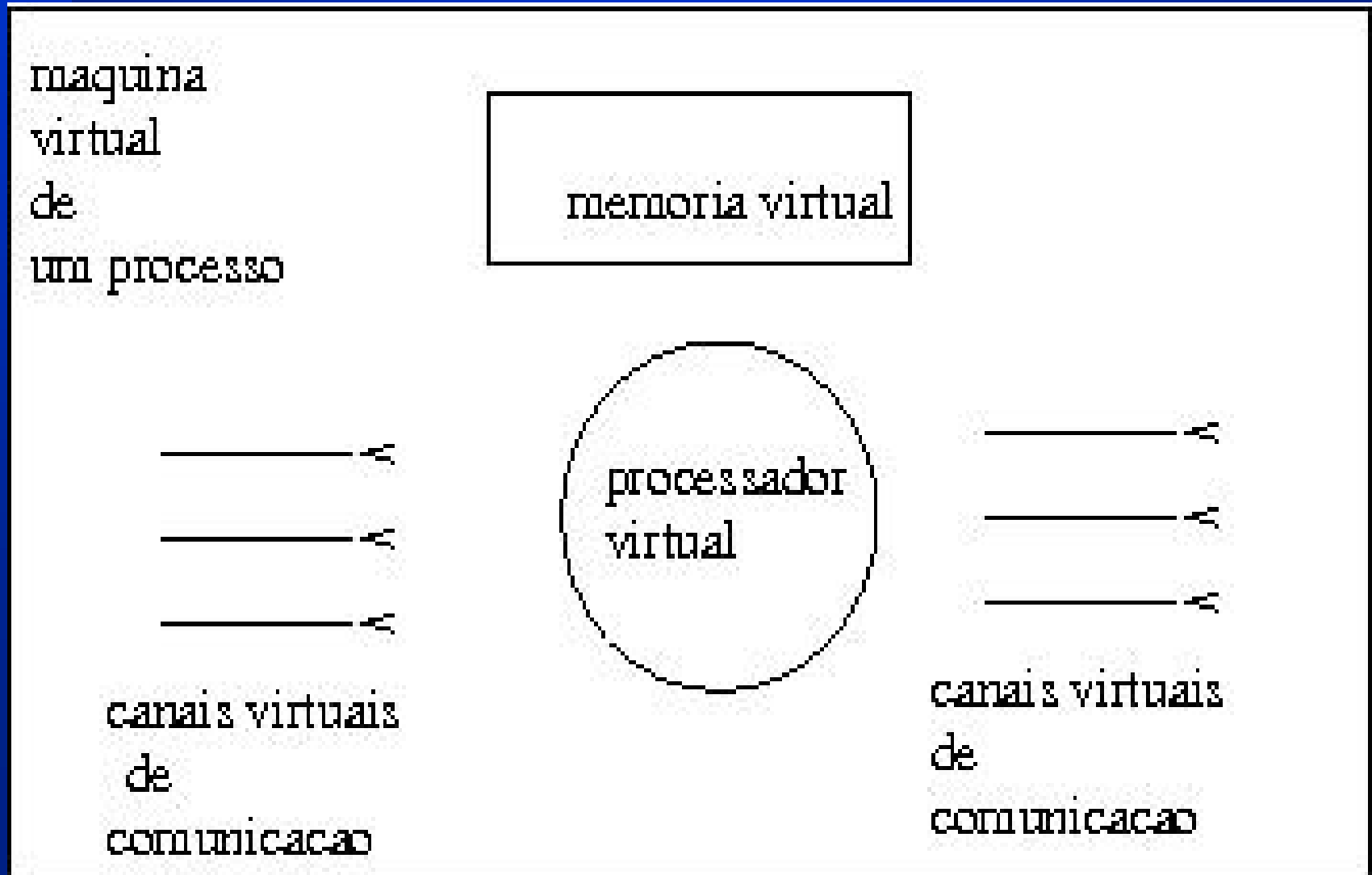
Processo = unidade de gestão criada pelo SO para controlar a execução de cada programa

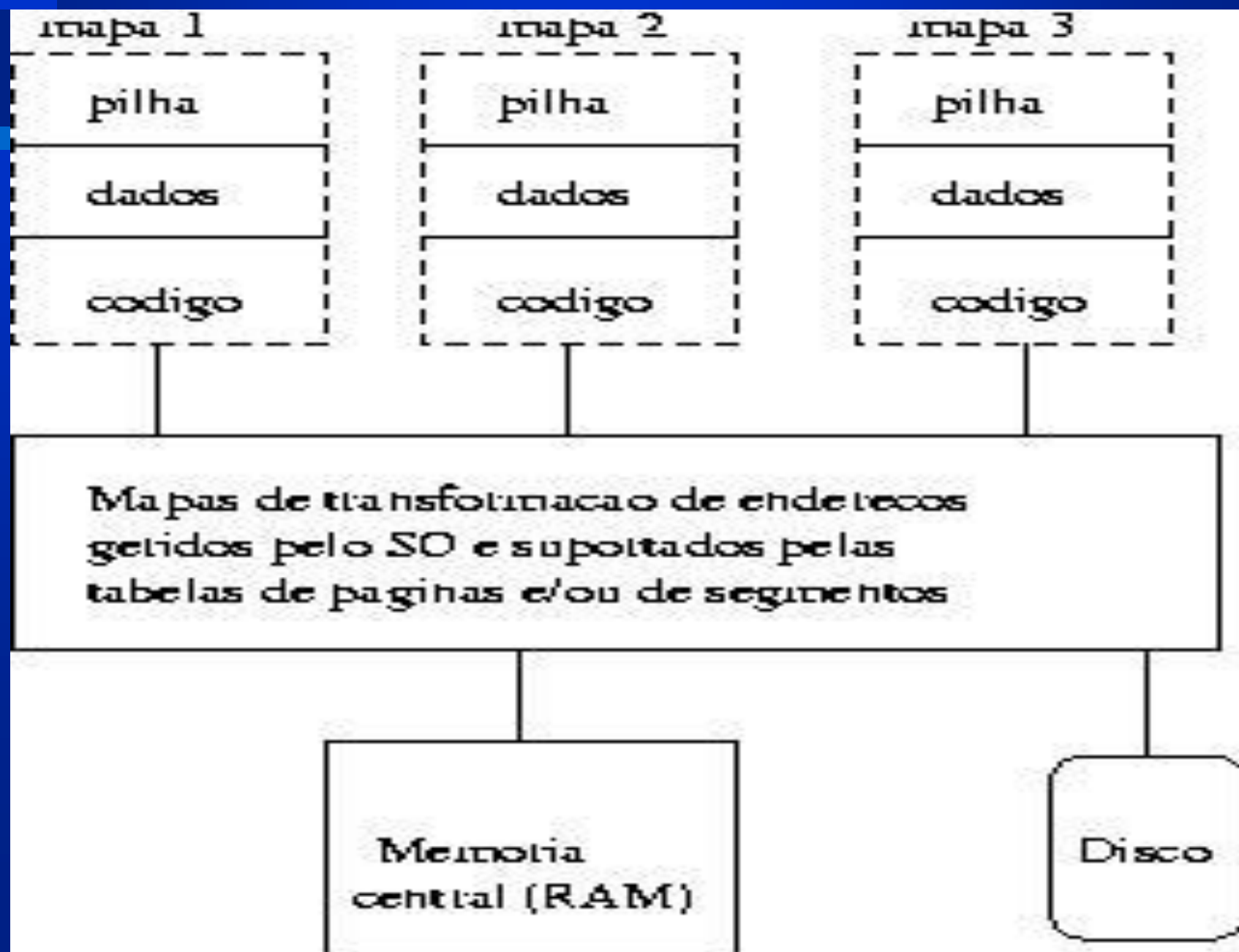


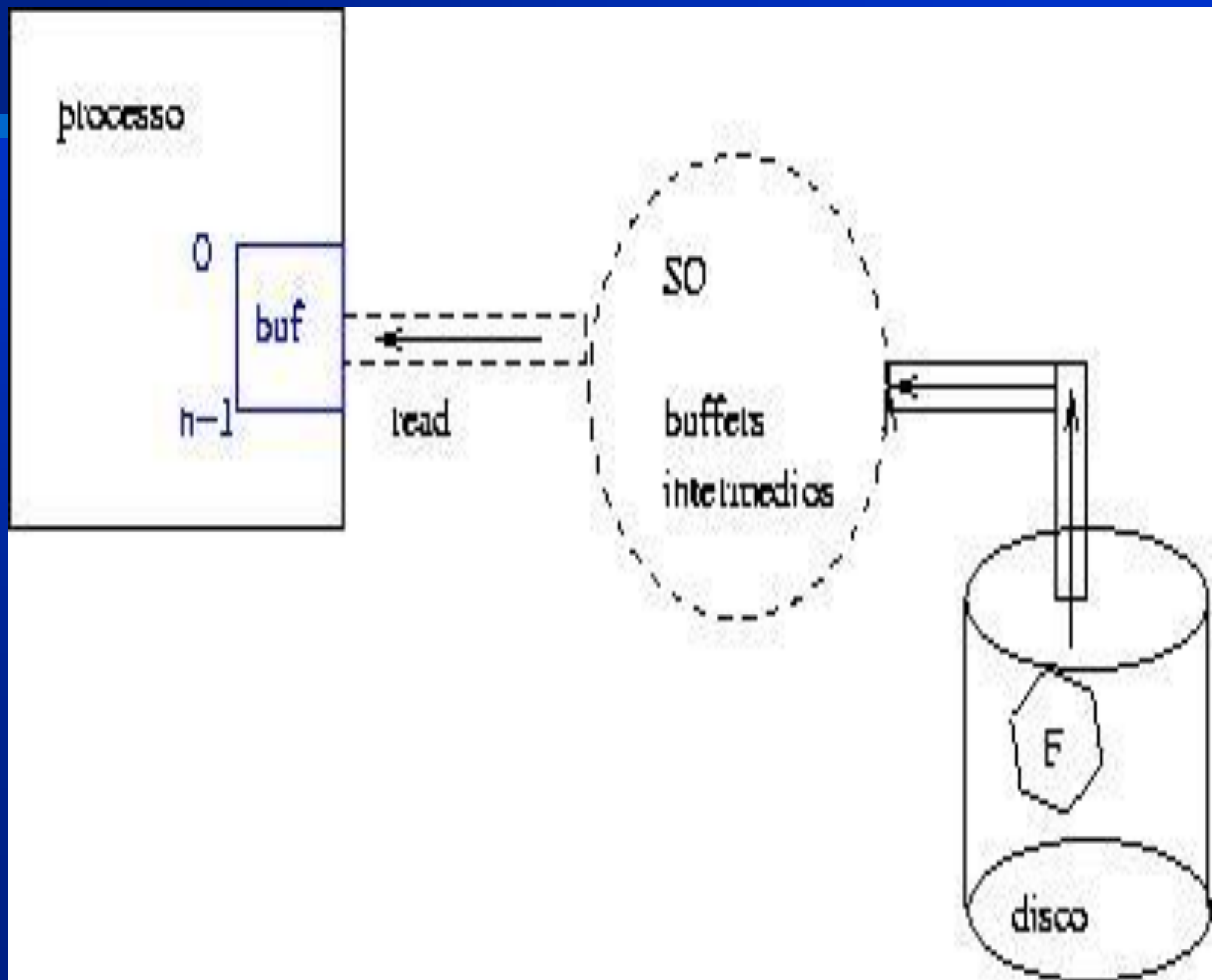
Ciclo de vida de um processo



Máquina virtual de cada processo







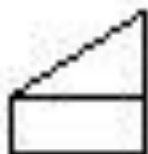
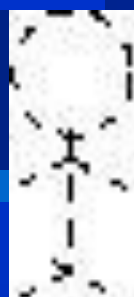
U1

U2

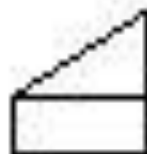
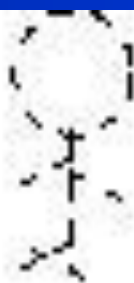
P1

P2

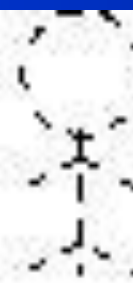
SO



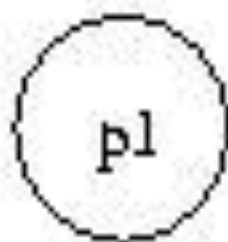
t1



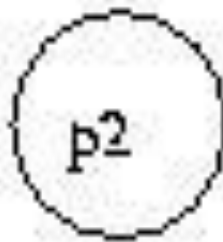
t2



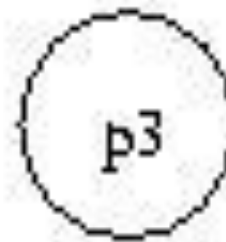
t3



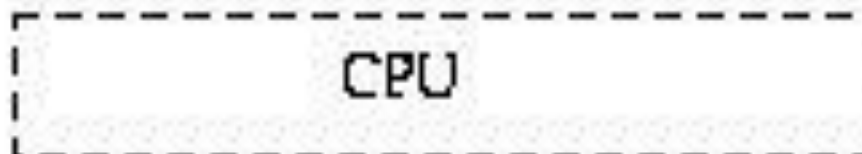
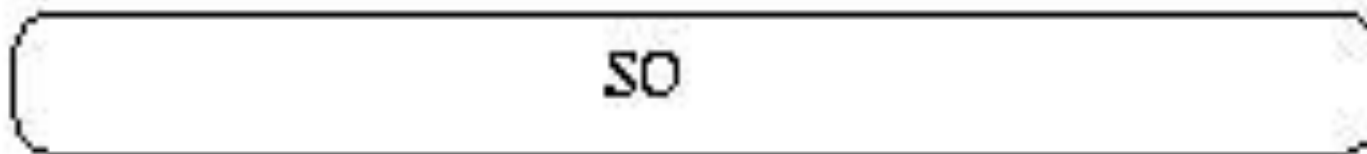
interpretador
de comandos



interpretador
de comandos



interpretador
de comandos



Multiprogramação

- Exemplo: 4 processos **concorrem** por 1 CPU
- Se um processo espera por I/O ou termina, o SO atribui o CPU a outro processo

