

Sistemas Operativos

Licenciatura Engenharia Informática
Mest. Int. Engenharia Computacional

Ano letivo 2018/2019

Nuno Lau (nunolau@ua.pt)

Sistemas Operativos (2018/2019)

Escolaridade:

2h TP / semana 2h P / semana 1h OT/ semana

Docentes (aulas TP e P):

Nuno Lau

nunolau@ua.pt

IEETA (IRIS Lab / 2.07)

Página web em elearning.ua.pt

login: utilizador universal

Slides adaptados dos usados em edições anteriores da disciplina
(Prof. António Rui Borges) e na bibliografia

- Apresentar os conceitos mais importantes sobre a organização dos sistemas operativos atuais numa perspetiva funcional
- Introduzir o ambiente de interação com o sistema computacional baseado no processamento de linha de comando
- Apresentar o sistema operativo como uma abstração que fornece ao programador de aplicações um modelo de máquina virtual baseado em *chamadas ao sistema*
- Introduzir a programação concorrente e os mecanismos principais de comunicação e de sincronização entre processos;
- Familiarizar os alunos com o interface de interação fornecido pelo Unix.

- Compreensão do mecanismo da multiprogramação e da organização geral de um sistema operativo;
- Capacidade de realização de tarefas administrativas simples para configuração e gestão do sistema operativo;
- Capacidade de desenvolvimento de pequenas aplicações que tiram partido das APIs fornecidas pelo modelo de máquina virtual do sistema operativo, tendo em vista promover a robustez e a portabilidade de código
- Capacidade de projeto de aplicações concorrentes simples

Programa (aulas teorico-práticas)

1. Conceitos Introdutórios
2. Ambiente de interação de linha de comando
3. Gestão do Processador em Multiprogramação
4. Comunicação entre Processos
5. Gestão da memória
6. Input / Output
7. Sistema de ficheiros
8. Proteção e Segurança

As aulas práticas seguem uma filosofia do *saber fazer* e visam a realização de pequenos trabalhos distribuídos por grupos de aulas.

Grupo 1 – Processamento da linha de comando

- Construção de pequenas tarefas para configuração e gestão do ambiente de interação apresentado pelo ambiente Unix.
- Resolução de um problema proposto.

Grupo 2 – Modelo de máquina virtual baseado em *chamadas ao sistema*

- Construção de pequenas aplicações em linguagem C que promovem a comunicação com os recursos do sistema computacional em ambiente Unix.
- Miniteste

Grupo 3 – Programação concorrente

- Construção de pequenas aplicações concorrentes usando os mecanismos de comunicação e sincronização mais comuns em ambiente Unix: semáforos e memória partilhada, passagem de mensagens e pipes.
- Resolução de um problema proposto.

- Cada grupo de trabalho tem 2 elementos
- O grupo tem que efetuar uma implementação precisa dos problemas propostos
- O trabalho será avaliado essencialmente de acordo com as funcionalidades que apresenta
- Os trabalhos são desenvolvidos fora das aulas práticas
- A nota atribuída aos trabalhos efetuados não será necessariamente a mesma para todos os elementos do grupo
- **O plágio será fortemente penalizado**

Frequência das aulas P é obrigatória

O estudante que faltar injustificadamente a mais de 20% das aulas com componente prática:

- reprova automaticamente à respetiva unidade curricular, ficando impedido de apresentar-se a qualquer prova da mesma durante o respetivo ano lectivo.

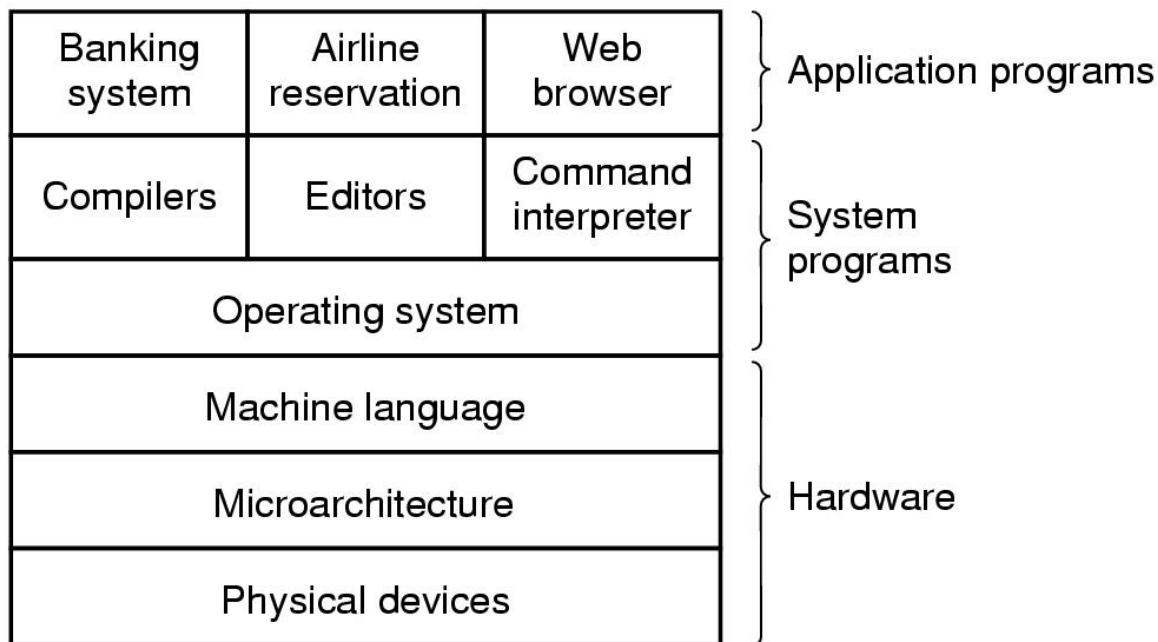
- Componente TeoricoPrática
 - Teste final (durante época de exames)
 - Nota mínima: 8
- Componente Prática
 - 2 trabalhos
 - Mini-teste (20 de Novembro?)
 - $30\%.TP1 + 30\%.TP2 + 40\%*MT$
 - Nota mínima: 8
- Nota final
 - $NF = 50\%.CTP + 50\%.CP$

- *Operating System Concepts with Java*, Silberschatz, Galvin, Gagne, 8th edition, Wiley, 2009
- *Operating System Concepts*, Abraham Silberschatz , Peter Baer Galvin, Greg Gagne, 9th edition, Wiley, 2012
- *Modern Operating Systems*, Andrew S. Tanenbaum, 4th edition, Pearson, 2014
- *Operating Systems, Principles and Practices*, Anderson and Dahlin, 2nd edition, Recursive Books, 2014
- *Sistemas Operativos*, José Alves Marques et al. 2^a edição, FCA, 2012

Questões

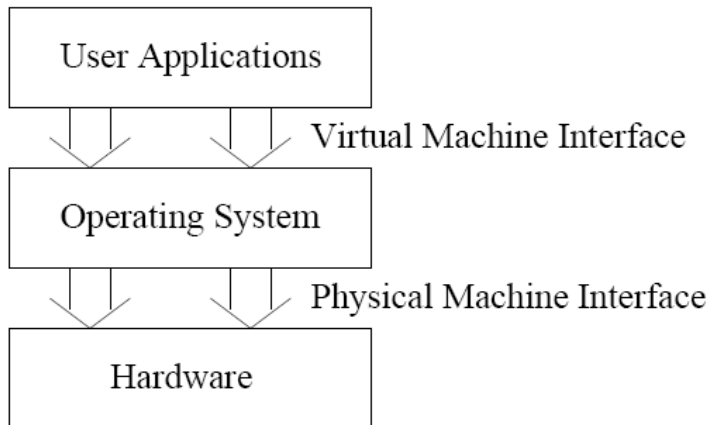
O que é um Sistema Operativo?

- O Sistema Operativo é o programa base que estabelece a interface entre os programas de aplicação e o hardware.



Objectivos do Sistema Operativo

- Executar os programas de aplicação
- Tornar o hardware mais fácil de usar
 - O SO cria um nível de abstracção que esconde muitos dos pormenores da utilização de dispositivos específicos (usando *device drivers*)
- Usar o hardware de forma eficiente
 - O SO gere os recursos de hardware do sistema de forma a tornar a sua utilização mais eficiente, justa e segura

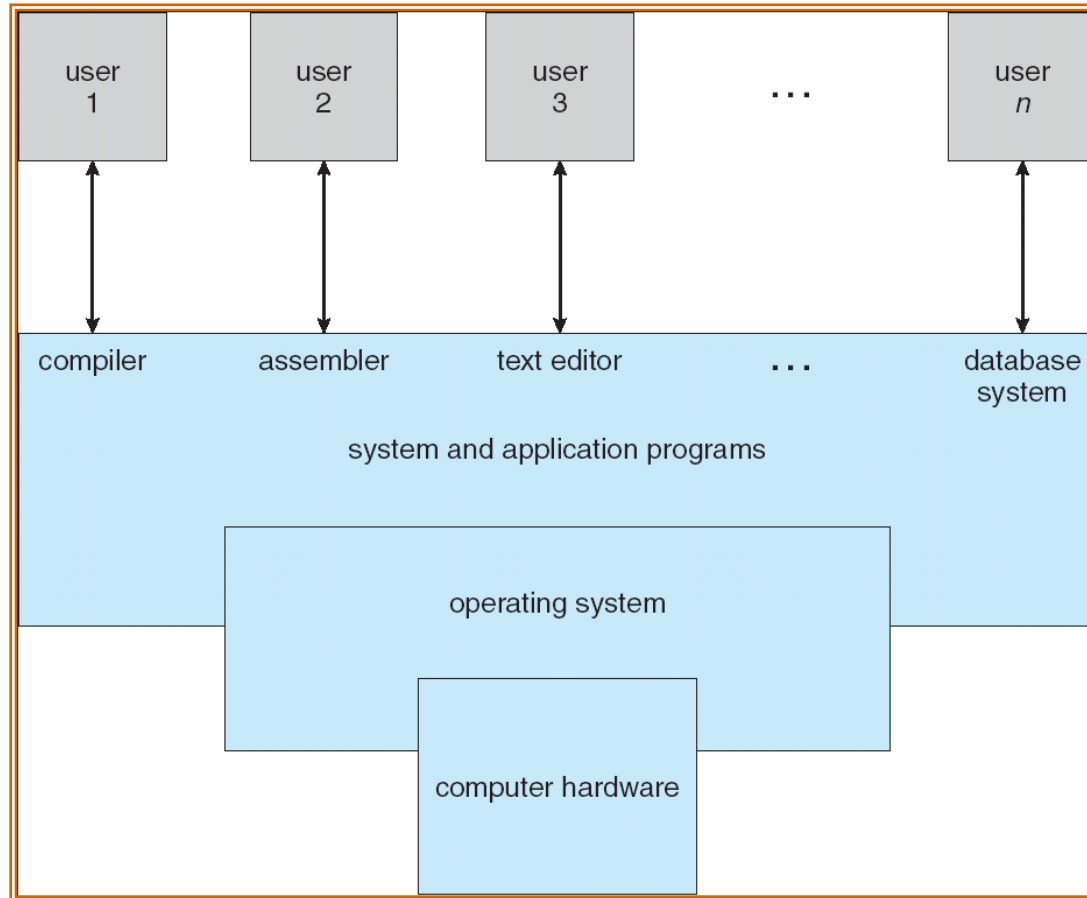


Os 2 últimos objectivos podem facilmente entrar em conflito

Um sistema computacional pode ser dividido em 4 componentes:

- Hardware
 - CPU, Memória, Dispositivos I/O
- Sistema Operativo
 - Controla e coordena o uso de hardware entre as várias aplicações e utilizadores
- Programas de aplicação
 - Processadores de texto, compiladores, browsers, bases de dados, jogos, etc
- Utilizadores
 - Pessoas, máquinas, outros computadores

Sistema Computacional



- **Serviços:** O SO cria serviços standard que são implementados pelo hardware
 - Exemplos: Sistema de ficheiros, memória virtual, redes, etc
 - Sistema operativo como criador de máquina virtual
- **Coordenação:** O SO coordena várias aplicações e utilizadores de modo a garantir segurança, eficiência e justiça na utilização dos recursos
 - Exemplos: concorrência, protecção da memória, segurança
 - Sistema operativo como gestor de recursos
- **Controlo:** O SO controla a execução dos programas prevenindo erros e uso impróprio do computador
 - Exemplos: escalonamento do CPU, criação de novos processos, seg fault, etc
- **Objectivo:** Criar um SO que é simultaneamente fácil de usar e eficiente

- Estabelecimento do ambiente de base de interação com o utilizador
- Mecanismos de execução controlada de programas
- Mecanismos de comunicação entre programas e respetiva sincronização
- Disponibilização de facilidades para o desenvolvimento, teste e depuração de programas
- Espaço de endereçamento virtual dos programas é independente das limitações da memória física
- Sistemas de ficheiros
- Modelo geral de acesso a dispositivos de I/O
- Detecção de situações de erro

- Concorrência
 - Permite que vários programas sejam executados em simultâneo
 - Também vários utilizadores em simultâneo
- Dispositivos de I/O
 - CPU continua a trabalhar enquanto I/O não responde
 - Mecanismos comuns para acesso a vários tipos de dispositivos
- Gestão da memória
 - SO gere as alocações de memória e transferências de dados entre memória e disco
- Ficheiros
 - Espaço em disco é organizado num sistema de ficheiros capaz de armazenar vários ficheiros de tamanho variável
- Sistemas distribuídos e redes
 - Permite que um grupo de computadores trabalhem de forma conjunta para resolver um problema

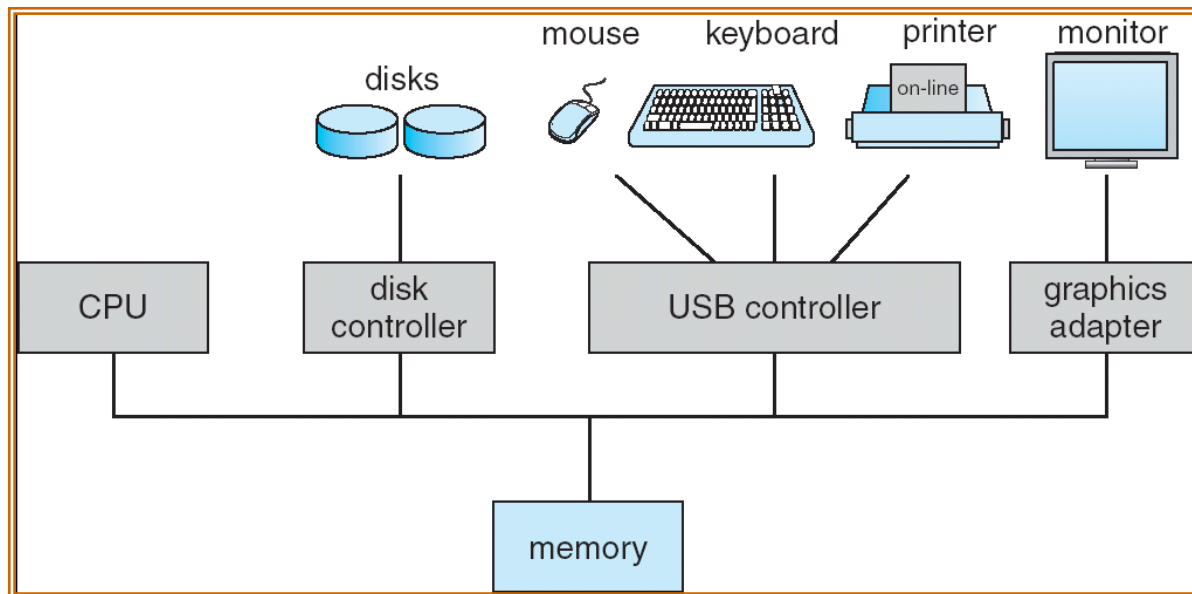
Tipos de Sistemas Operativos

- Sistemas Operativos para Mainframes
 - Serviços: Batch, Transações e Timesharing
 - Ex: OS/390
- Sistemas Operativos para Servidores
 - Partilha de recursos de hardware e software
 - Ex: Solaris, FreeBSD, Linux, Windows Server 201x
- Sistemas Operativos para Multiprocessadores
 - Ex: Windows, Linux
- Sistemas Operativos para PCs
 - Ex: Windows, Linux
- Sistemas Operativos para Dispositivos Móveis
 - Ex: iOS, Android
- Sistemas Operativos para Sistemas Embebidos
 - Ex: QNX, VxWorks
- Sistemas Operativos para Nós Sensoriais
 - Ex: TinyOS
- Sistemas Operativos de Tempo Real
- Sistemas Operativos para Smart Cards

- Programa de *bootstrap* é carregado quando o computador arranca ou é reinicializado
 - Tipicamente armazenado em ROM ou EPROM (*firmware*)
 - Inicializa vários dispositivos do sistema
 - Carrega o núcleo (*kernel*) do sistema operativo e começa a sua execução

Organização do computador

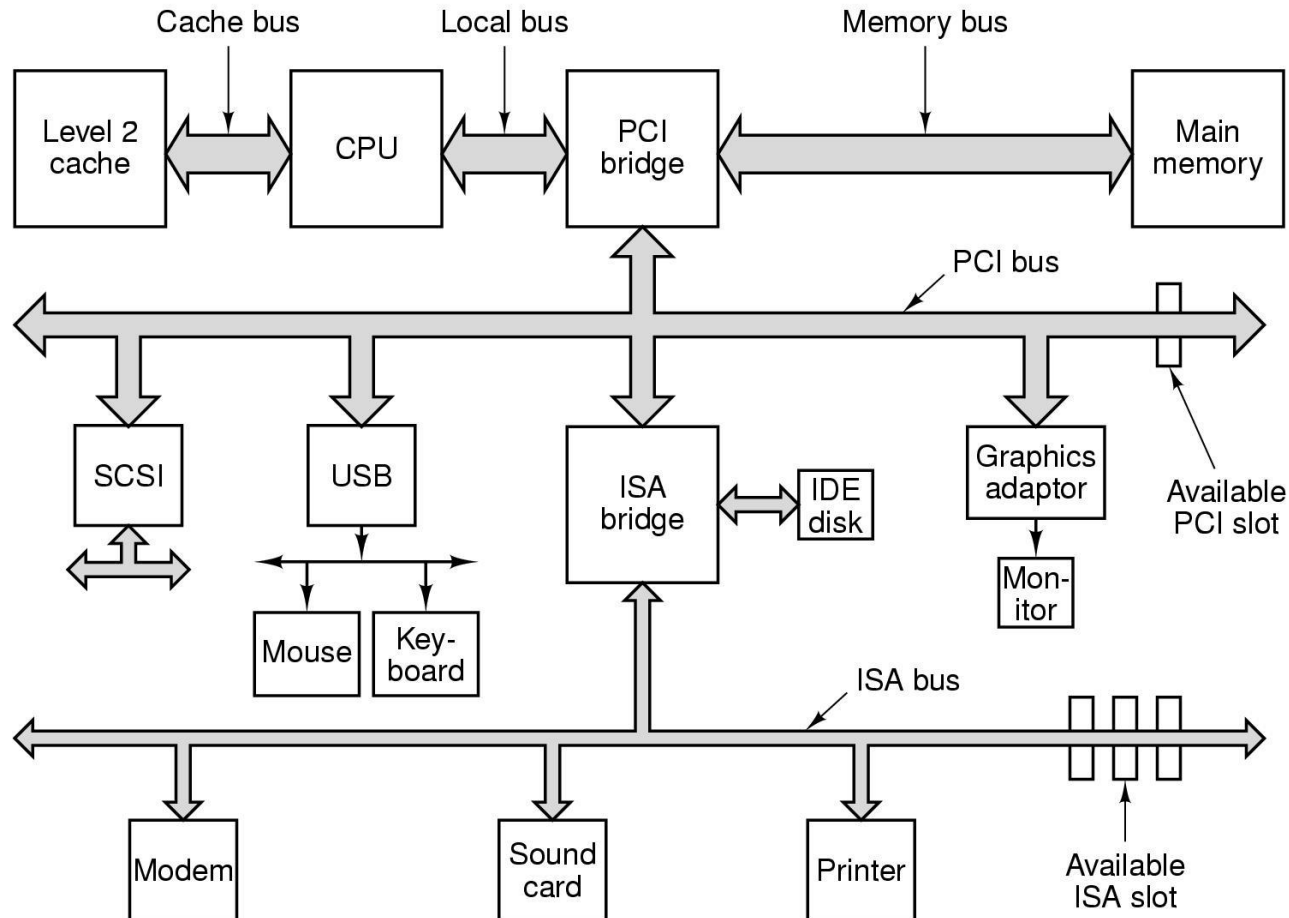
- Um ou mais CPUs e controladores de dispositivos ligados à memória através de barramento
- Execução concorrente de CPU e dispositivos origina conflitos no acesso à memória



Organização do computador

- CPUs e controladores de dispositivos de I/O executam em paralelo
- Cada controlador de dispositivo trata um tipo particular
- Controladores de dispositivo têm *buffer* local
- CPU move dados de/para memória e de/para *buffers* locais
- Transferências de I/O são do dispositivo para o *buffer* local do respectivo controlador e depois para a memória
- Controlador do dispositivo informa CPU que terminou a operação através do envio de uma interrupção

Organização do computador



- Visualizar aplicações em execução
 - Windows
 - Task Manager/Gestor de Tarefas
 - Linux
 - Comandos: `ps`, `top`, `htop`