



Sistemas Operativos

2021 / 2022

leic-so-alameda@disciplinas.tecnico.ulisboa.pt
leic-so-tagus@disciplinas.tecnico.ulisboa.pt
leti-so-tagus@disciplinas.tecnico.ulisboa.pt

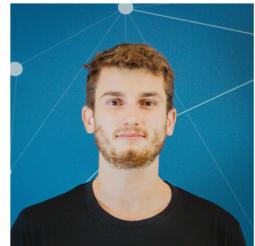
Corpo Docente

- Paolo Romano (responsável Alameda)
- André Breda
- Bernardo Conde
- Daniel Matos
- Daniel Reis
- Gonçalo Raposo
- Helena Cruz
- Henrique Guerra
- João Martinho
- Pedro Moreira



Corpo Docente

- João Barreto (responsável Tagus)
- André Ferreira
- Baltasar Dinis
- Daniel Reis
- Diogo Barbosa
- Marcelo Mouta



Que rumores já ouviram sobre SO?



“Porquê SO,
se eu provavelmente nunca
trabalharei dentro de um
sistema operativo?”

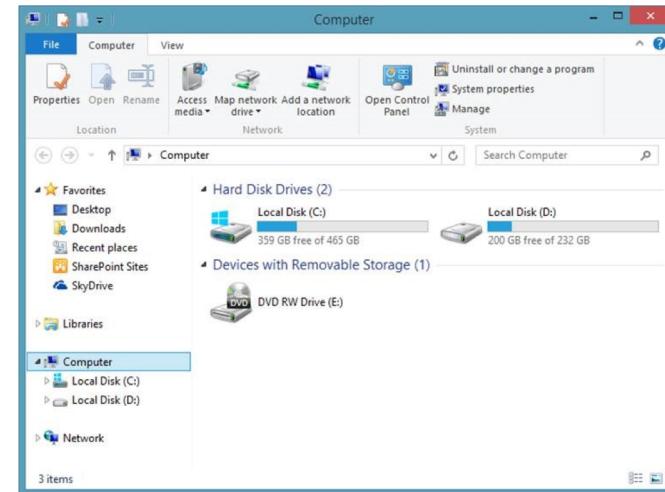
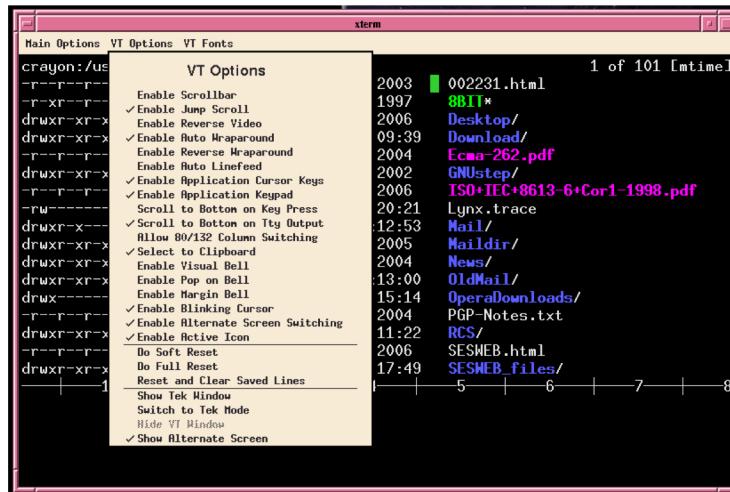


O que é usar um computador ?

PRIMEIRA RAZÃO

O SO define funcionalmente o uso do computador

- É o SO que caracteriza a “máquina Informática” que usamos
- A “operação” baseia-se numa interface que a o SO disponibiliza



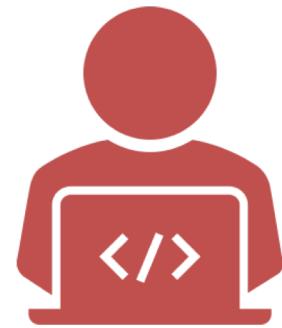
SEGUNDA RAZÃO

O Sistema Operativo cria uma máquina virtual totalmente independente do hardware onde se executa

Mais que interfaces, o SO fornece uma máquina virtual

- Lembram-se do computador estudado em IAC ou AC?
- O Sistema Operativo virtualiza praticamente todos os mecanismos de hardware e cria **uma nova máquina**

Uma nova máquina (virtual)?



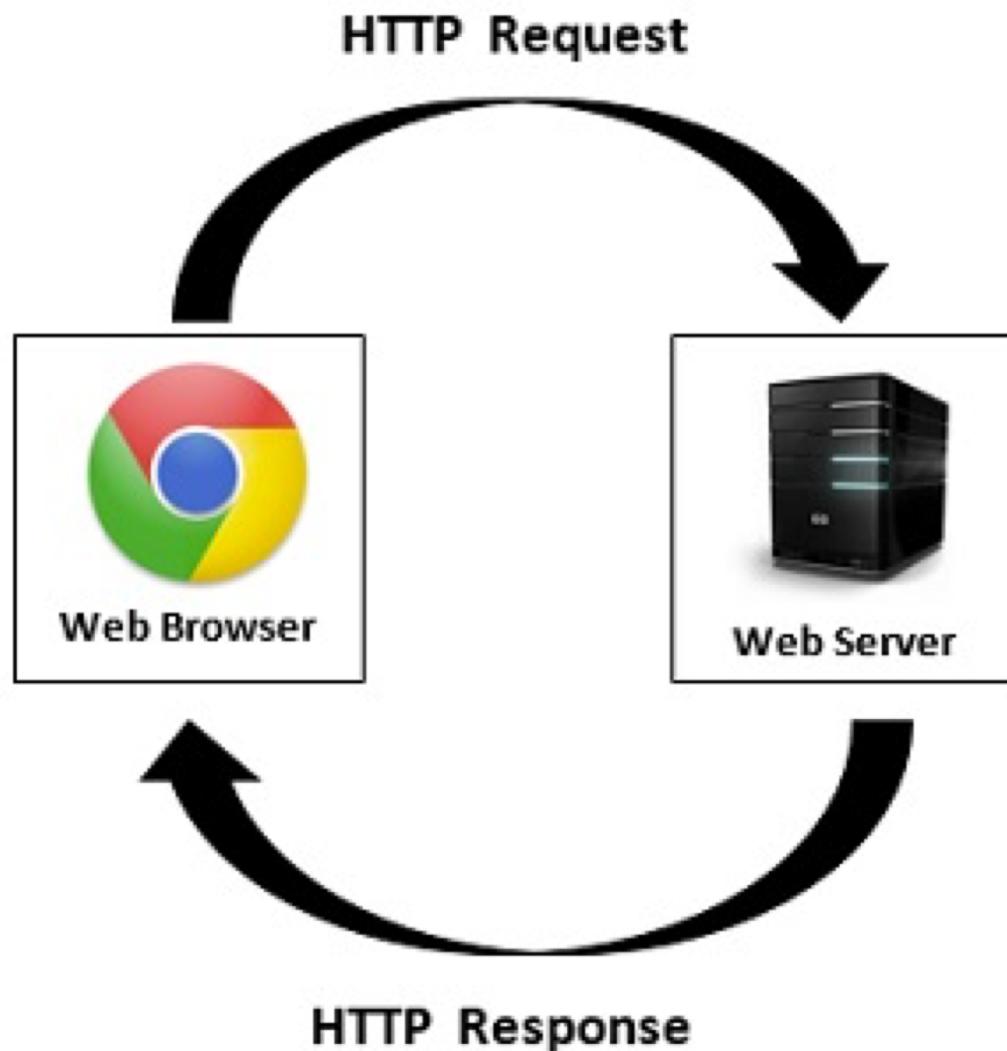
Para que serve um Sistema Operativo?

- Gerir Recursos
 - Seria difícil às aplicações controlar todos os aspectos da máquina física (interrupções, organização da memória, dispositivos, ...)
 - Abstrair os **recursos físicos**, oferecendo às aplicações um conjunto de **recursos lógicos**
- Exemplos?

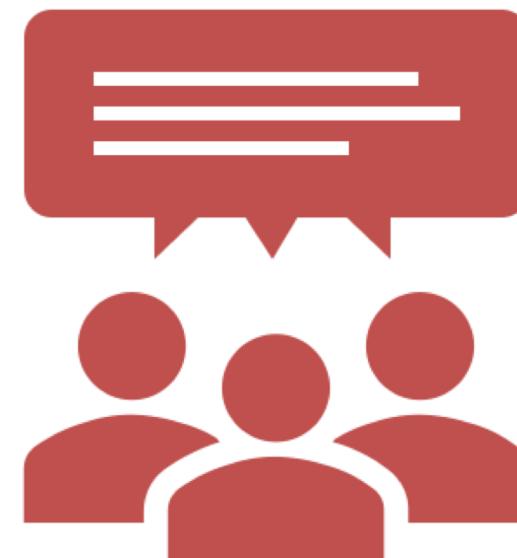
Recursos lógicos vs. físicos

Recursos Lógicos	Recursos Físicos virtualizados
Processos	CPU
Espaços de endereçamento virtuais	Memória RAM, Unidade de Gestão de Memória
Ficheiros	Discos e dispositivo de memória de massa
Periféricos virtuais	Periféricos físicos
Comunicação entre Processos	Partilha de memória, redes de dados
Utilizadores, permissões, privilégios	Mecanismos de segurança físicos do hardware

Exemplo: browser e servidor web

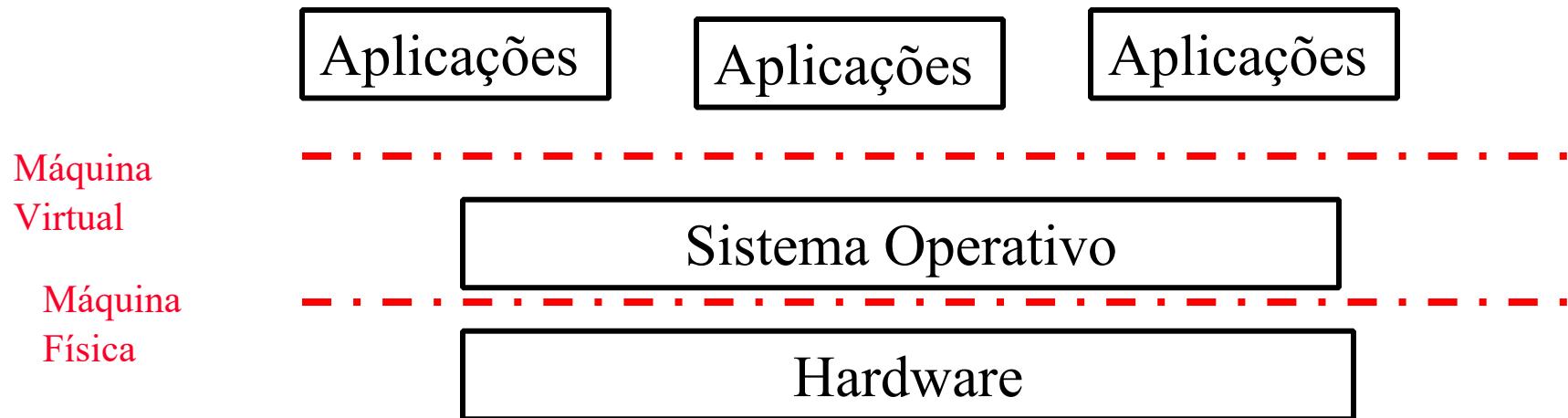


- De que formas é que o SO suporta uma aplicação web?



- As aplicações com os seus dados são o que realmente tem **valor** para as organizações
- As grandes vantagens de ter uma aplicações que “corre” em Unix/Linux/Windows/MacOS:
 - Temos a certeza de a conseguir manter e fazer evoluir neste SO
 - O SO cria uma máquina virtual **segura, fiável e otimizada**

Missão do Sistema Operativo



- Criar uma máquina virtual sobre a máquina física que ofereça os recursos lógicos básicos necessários ao desenvolvimento das aplicações
- Independente do hardware onde se executa



A possibilidade de usar o Sistema operativo para estender as capacidades das linguagens de programação

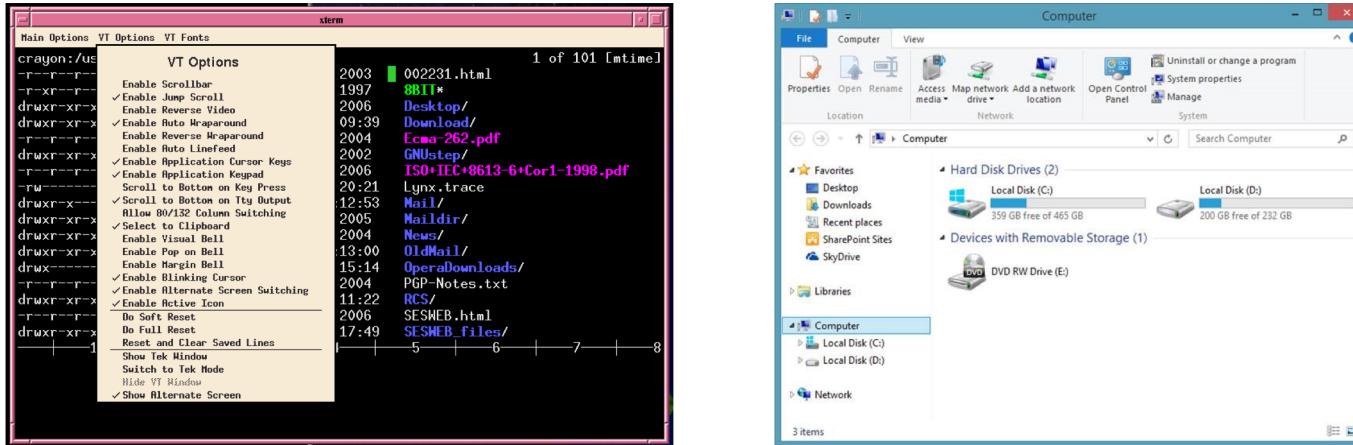
TERCEIRA RAZÃO

Ir além da linguagem de programação

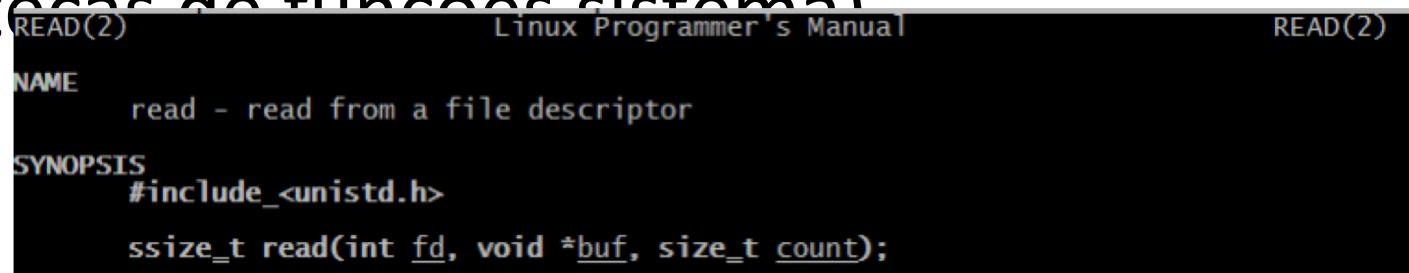
- Os sistemas disponibilizam uma interface que permite a qualquer programador estender o seu ambiente de programação permitindo-lhe criar aplicações muito sofisticadas
 - Paralelismo
 - Optimização da memória
 - Persistência
 - Comunicação entre processos, local ou distribuída
 - Tratamento de erros
 - Novos periféricos, protocolos, etc.

“Interfaces” do Sistema Operativo?

1. Interface Operacional



1. Interface programática (bibliotecas de funções sistemáticas)



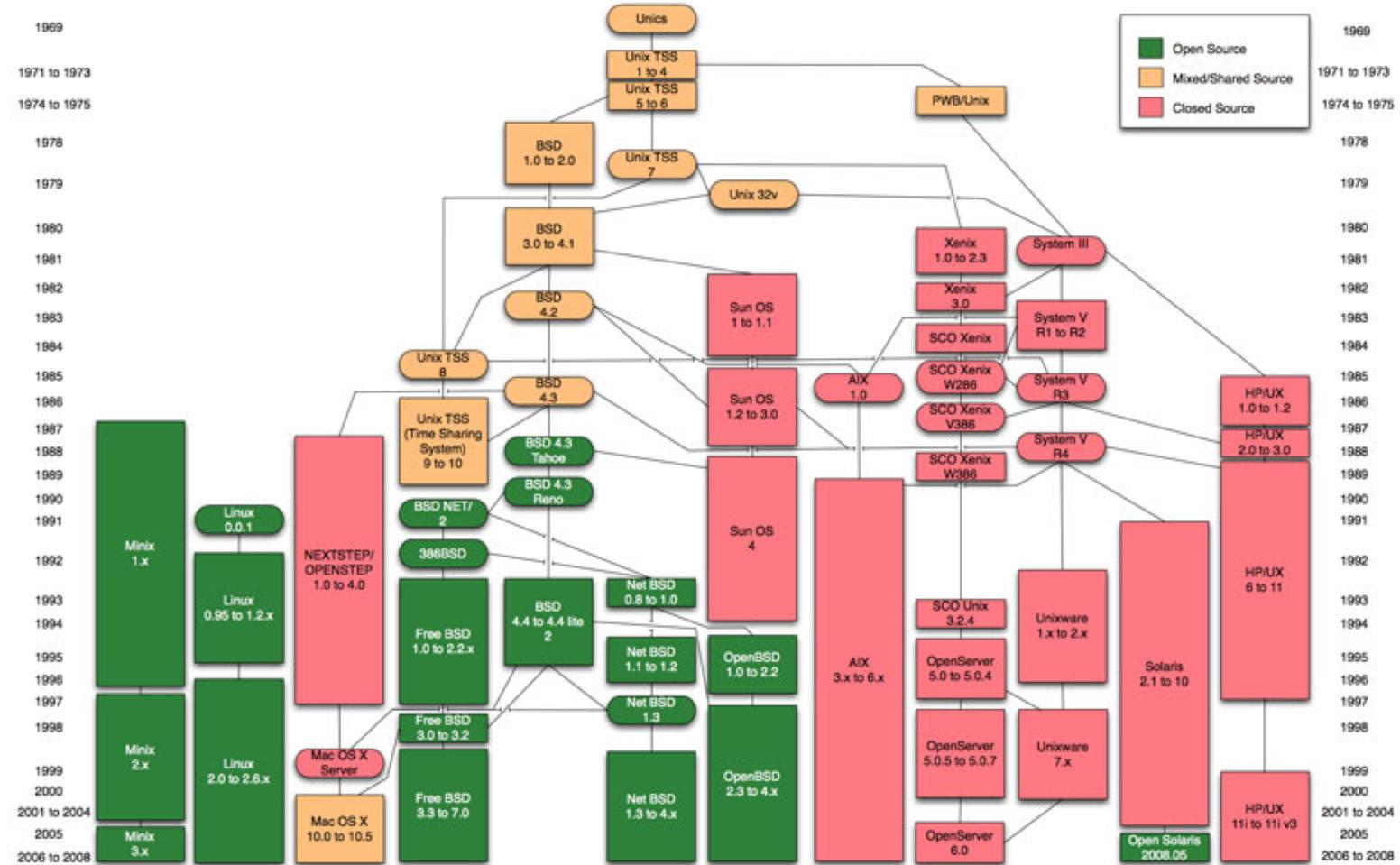


OS SISTEMAS OPERATIVOS DE REFERÊNCIA

Que SOs já usaram hoje?

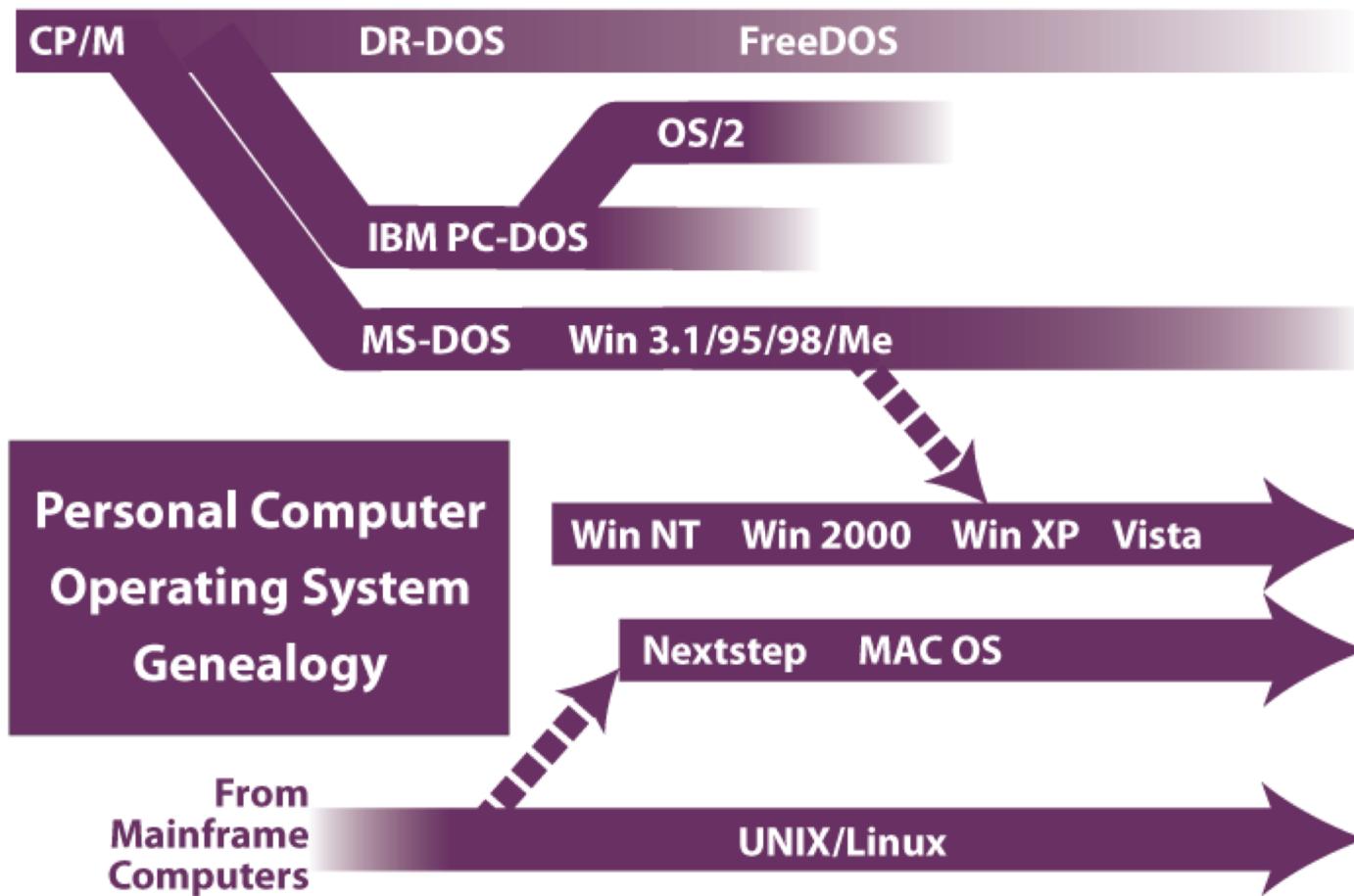


A família Unix



Créditos do diagrama:
Eraserhead1

Outras famílias



E as novas gerações





QUAL O MELHOR SO ATUAL?

Que aspecto valorizam mais quando escolhem um SO?



Critérios de Qualidade do SO

- Desempenho
 - Gestão eficiente dos recursos físicos que suportam os recursos lógicos
 - Tempo de resposta, Débito, Previsibilidade, Justiça
- Segurança.
 - Isolamento dos Utilizadores
 - Permitir partilha segura de recursos lógicos
- Fiabilidade e Disponibilidade
 - Detectar um conjunto de faltas
 - Tolerar um conjunto de erros
- Interface de programação completa e simples.
 - Facilitar a concepção das aplicações, a sua manutenção e portabilidade
- Interface de operação e gestão dos recursos lógicos fácil de utilizar
- Portabilidade
 - das aplicações e do próprio SO
- Adoção



QUAL O MELHOR SO PARA APRENDER?

Unix/Linux

- O Unix foi um sistema fundamental em toda a evolução da informática
 - O Unix é simples
 - Interfaces simples, centradas no conceito de ficheiro
 - O Unix e os seus programas utilitários estão escritos em C
 - Exemplo de como se consegue conceber um programa que com, muitas evoluções, vive há 4 décadas
- Vamos também usar Linux, uma evolução do UNIX
 - mas poderíamos usar outros descendentes



Objetivos da disciplina

Objectivos (1)

Conhecer os conceitos fundamentais subjacentes aos sistemas operativos, com ênfase nos seus mecanismos e algoritmos, assim como a estrutura interna dos sistemas operativos mais relevantes.

Objectivos (2)

Programar, ao nível sistema,
usando todas as funcionalidades dos
sistemas operativos
tendo em conta os modelos de
programação sequencial e concorrente.

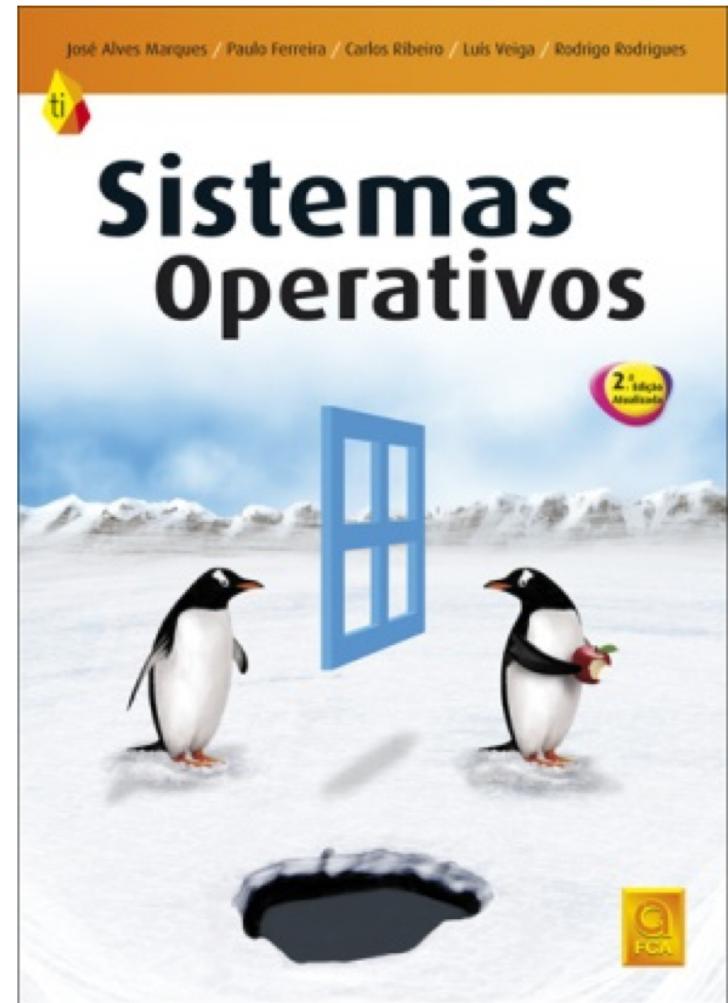
Ok, como é que isto vai
funcionar?

Aulas

- Aulas teóricas:
 - 3 aulas de 1 h + 1 aula de 2h / semana
- Aulas laboratório:
 - 2 aulas de 1,5 h / semana
 - Já começaram esta semana
- Todas as aulas são presenciais

Bibliografia

- Sistemas Operativos,
Editora FCA – 2^a Edição
- Alternativas sem
comprar:
 - biblioteca
 - pedir emprestado a
colegas
- 1^a Edição também serve



Método de Avaliação (1)

- A avaliação da disciplina tem 2 componentes:
 - teórica (50%)
 - projecto (50%).
- A nota mínima para aprovação à cadeira é de 10 (dez) valores.

Método de Avaliação (2)

- Componente teórica
 - 1 exame na época normal
 - 1 exame na época de recurso
 - substitui a nota do exame de época normal apenas caso a melhore
 - **Nota mínima da componente teórica: 9,0 valores**

Método de Avaliação (3)

- Componente Projeto
 - Grupos de 2 alunos
 - Formados na 1^a semana de cada turno laboratorial
 - Construído em 2 exercícios ao longo do semestre
 - P1 50% + P2 50%
 - Nota mínima de 9 valores
 - Nota do projeto do ano anterior é válida este ano se $\geq 10,0$
 - Presença obrigatória na discussão final
 - inclui teste prático: extensões ao código submetido

Método de Avaliação (5)

- Avaliação do projeto:
 - Cada exercício recebe nota preliminar, meramente **indicativa**
 - Nota final é obtida após a discussão
 - Realizadas **individualmente** após a última entrega
 - Caso normal: a nota final individual = nota preliminar do projeto
 - No entanto, a avaliação final pode reduzir arbitrariamente a nota preliminar

Datas

- Exames
 - Época normal, recurso: ver no fénix
- Projeto:
 - 1º enunciado: 3 Dezembro
 - Entrega 1ª parte/2º enunciado: 7 Jan., 23h59
 - Entrega 2º parte: 28 Janeiro, 23h59
 - Teste prático e discussão final: 31 Jan. a 2 Fev.

O programa num slide



1ª Parte

...com uma exceção:

- Sistemas de ficheiros
- introduzidos já na primeira semana
- Podem ser vistos como “quadro” do projeto de SO

2ª Parte



O programa num slide

Teóricas

Sistema de Ficheiros: introdução ao projeto

Programação usando o SO

Introdução à programação sistema

Programação com processos

Programação com tarefas e sincronização

Comunicação entre processos

Núcleo do SO

Gestão de processos, Memória Virtual, etc.

Labs

Ferramentas de desenvolvimento

P1

P2



Algumas dicas

Atenção

- O projecto exige **muito bom** domínio de C
 - Ter passado a IAED não chega
 - É preciso dominar a 100%:
 - a programação em C
 - o ambiente Unix/Linux
- Rever os slides de C avançado de IAED
- Instalar Linux na vossa máquina
 - Pode ser em vmware ou virtual box



QUIZ

<https://tinyurl.com/quiz-so>

Se não responderam certo a todas
as questões, então precisam rever a
matéria de IAED!

- Consultar slides no site de IAED
 - Incluindo aulas sobre ponteiros e IO
 - Tirar dúvidas nos horários de dúvidas de SO

Trabalhem os laboratórios a pensar na teoria

- Não deixem o colega de grupo tratar “daquela parte” do projeto
- Pair programming!
 - Youtube: “how to pair program”
- Estudem a teoria enquanto se dedicam ao projeto

LEIC/LETI – 2015/16 - 1º Exame de Sistemas Operativos
6 de Janeiro de 2016
Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas.
Duração: 2h30

Grupo I [5 Val]

Considerando os seguintes programas.

Prog1.c	Prog2.c
<pre>main () { int pid; int status; pid = fork (); if (pid == 0) { execl ("./prog2", "prog2", 0); /* ... */ } else if (pid > 0) { sleep (N); wait (&status); } else { /* ... */ } }</pre>	<pre>main () { for (i=0; i<900000000; i++); exit (5); }</pre>

Em cada uma das situações de erro. Para cada linha, descreva uma possível