Sistemas Operativos

Corpo Docente

Paolo Romano (Responsável) – romano@inesc-id.pt

Salvador Carvalhinho

L11,L13, salvador.carvalhinho@tecnico.ulisboa.pt

- Laura Cunha

L09, L12, laura.r.velasco.cunha@tecnico.ulisboa.pt

David Nunes

L03, L07, david.p.nunes@tecnico.ulisboa.pt

- André Noronha

L06, andre.peres.noronha@tecnico.ulisboa.pt

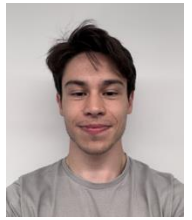
- Diogo Pacheco

L05, diogo.a.pacheco@tecnico.ulisboa.pt

- Miguel Pardal

L04, L08, miguel.pardal@tecnico.ulisboa.pt

- Pedro Ribeiro



L10, pedro.melo.ribeiro@tecnico.ulisboa.pt

Porquê estudar Sistemas Operativos?

- Os sistemas operativos começaram o seu desenvolvimento na década de 60 são dos primeiras grandes peças de software da arquitectura dos sistemas informáticos
- O Unix na sua versão inicial em C é de 1973 tem +50 anos!!

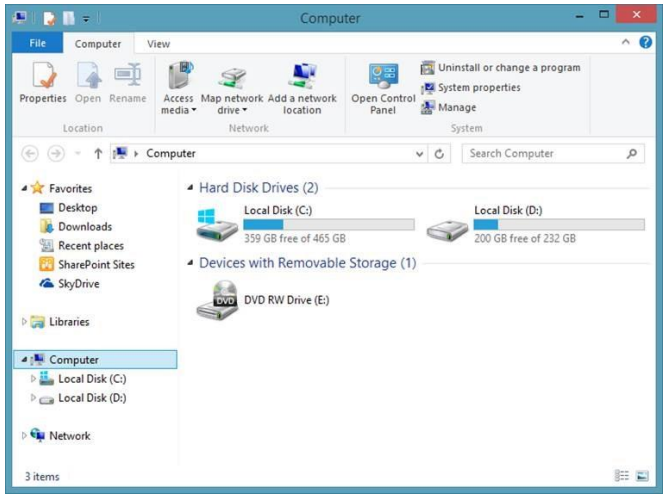
Qual a razão para que este tema justifique uma cadeira na maioria dos cursos de Engenharia Informática?

Como se utiliza um computador? Ou um Telemóvel

Primeira razão

O SO define funcionalmente o uso do computador

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



Fazer funcionar a máquina

- A interface já é um importante valor que justifica o nome inicial de Sistema Operativo: ***um programa que permitia operar a máquina***

- Mas o Sistema Operativo não facilita só a operação, faz mais, contrui uma nova máquina em cima do hardware

Uma máquina virtual independente do hardware onde se executa

Segunda Razão

O SO fornece uma máquina virtual

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

Recursos lógicos vs. físicos

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

Precisamos mesmo de um Sistema Operativo?

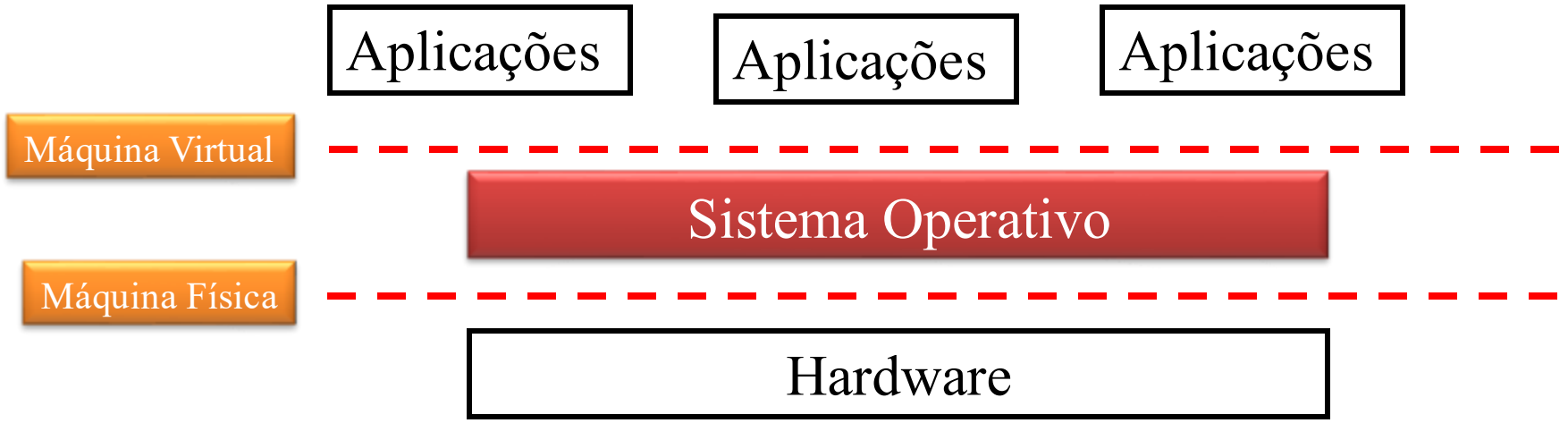
- As linguagens de programação poderiam produzir todo o código necessário para que um programa se executasse directamente sobre o hardware

- Desvantagens?
  - O esforço de programação seria muito grande
  - Um conjunto significativo de funções seria repetido
  - **Cada aplicação poderia optimizar o seu desempenho, mas globalmente a máquina ficaria subaproveitada.**
  - **Não seria possível ter politicas globais de segurança, tolerância a faltas, optimização**

Para que serve um SO?

- Definir e gerir um conjunto de Recursos Lógicos
- Abstrair os **recursos físicos**, oferecendo aos utilizadores um conjunto de **recursos lógicos, com uma interface simples de usar**
- Garantir que existe um politica de **segurança global** na gestão dos **recursos partilhados**

Sistema Operativo



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

Valor do Sistema Operativo

- As aplicações com os seus dados são o que realmente tem **valor** para as organizações. É a razão de investimentos em tecnologias de informação e no limite de se formarem engenheiros informáticos

- A grande vantagem de ter aplicações que “correm” em Unix/Linux/Windows/IOS é que temos a certeza de as conseguir manter e fazer evoluir neste sistema independentemente das inúmeras arquiteturas de hardware ou do uso de *cloud*

Uma máquina virtual segura para execução das aplicações

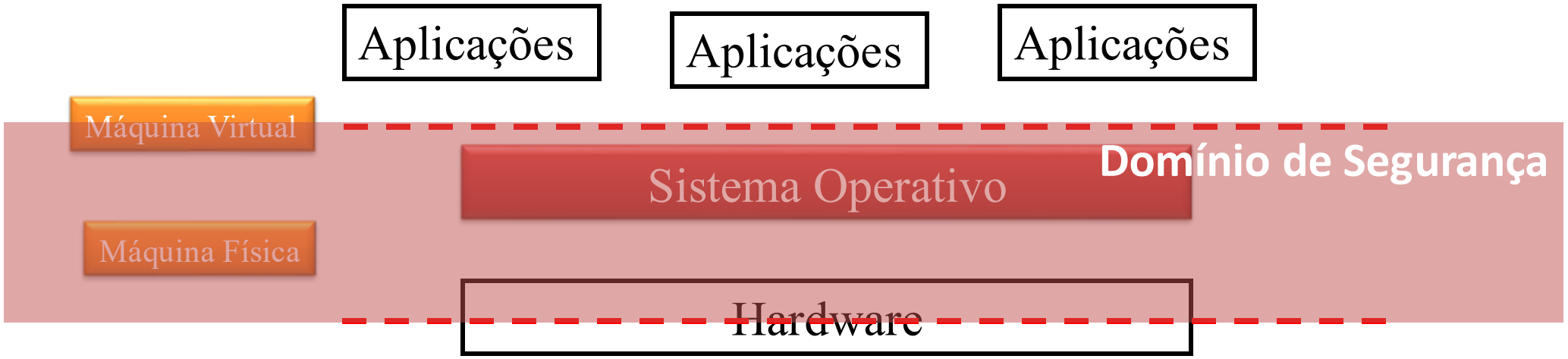
Terceira Razão

Deixar as aplicações usar diretamente o hardware?

Aplicações executam-se em modo não privilegiado

- Aplicações como crianças a brincar em “caixas de areia”
  - Aceder às suas variáveis, executar operações aritméticas, ler/escrever ficheiros, etc.
  - Proibidas de executar operações “perigosas” sobre os recursos físicos

Sistema Operativo



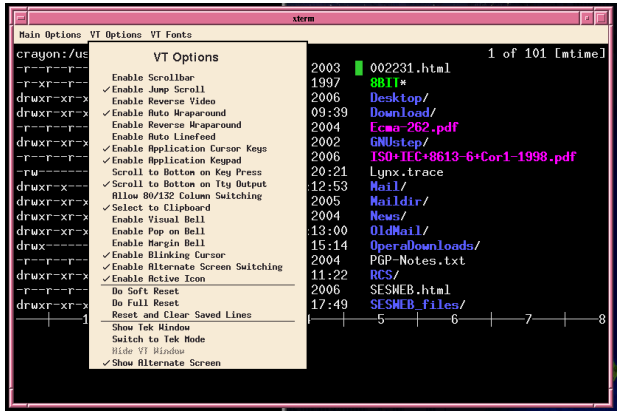
O Sistema Operativo cria uma máquina segura (dentro de alguns condicionalismos….).
Hoje em dia a maioria das arquiteturas de segurança dos sistemas informáticos têm como base esta pedra basilar

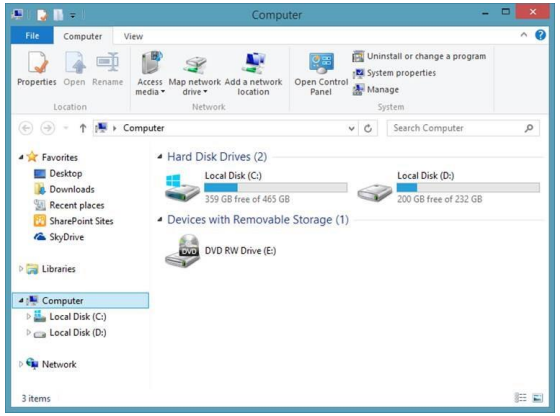
O Sistema Operativo estende as capacidades de programação

Quarta Razão

“Interfaces” dos recursos do Sistema Operativo?

1. Interface operacional





2. Interface programática - bibliotecas de funções sistema

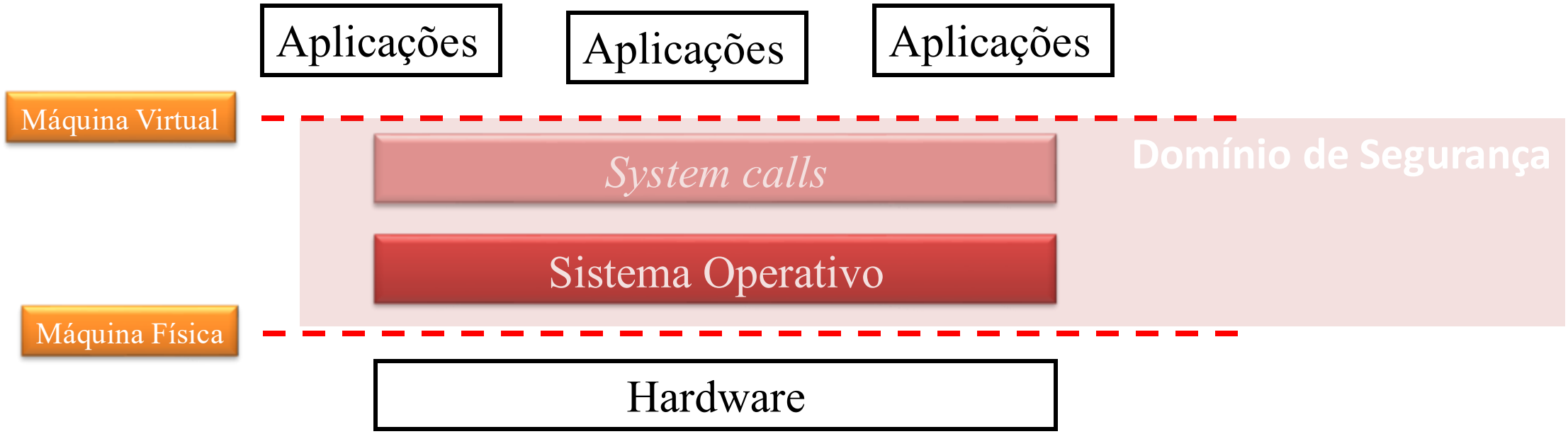
```
READ(2)                    Linux Programmer's Manual                    READ(2)

NAME
       read - read from a file descriptor

SYNOPSIS
       #include <unistd.h>

       ssize_t read(int fd, void *buf, size_t count);
```

A interface programática do SO

- Os sistemas disponibilizam uma interface (*system calls*) que permite aos programadores estenderem o seu ambiente de programação permitindo-lhe criar aplicações muito sofisticadas, incluindo, explorando:
  - Paralelismo
  - Comunicação entre processos local ou distribuída
  - Optimização da memória
  - Tratamento de erros
  - Novos periféricos, protocolos, etc.

- Hoje em dia muitas destas possibilidades forem incluídas nos ambientes de programação, mas baseiam-se neste conceitos originais dos Sistemas Operativos.

Sistema Operativo



- Criar uma máquina virtual sobre a máquina física que ofereça os recursos lógicos necessários ao desenvolvimento das aplicações
- Independente do hardware onde se executa
- Otimizada no desempenho
- Segura e confiável

Conclusão

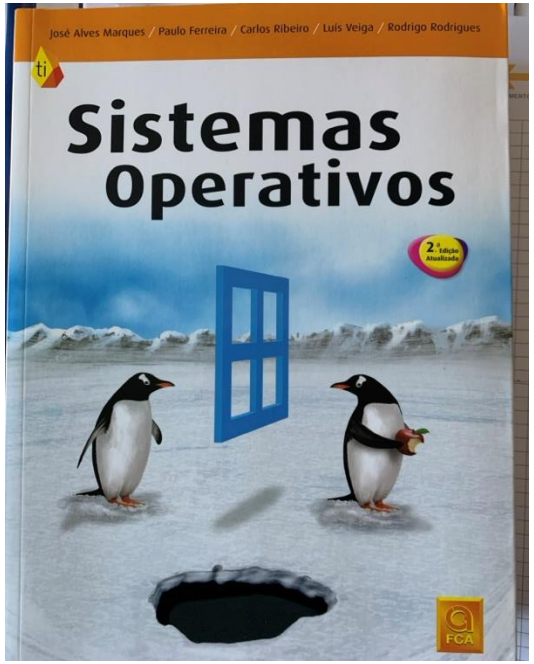
- Objetivos do sistema Operativo
  - Suportar eficientemente as aplicações
  - Garantir segurança e fiabilidade das operações
  - Garantir que as aplicações não são afectadas pela mudança de hardware e configuração
  - Permite estender o modelo de programação das linguagens de programação

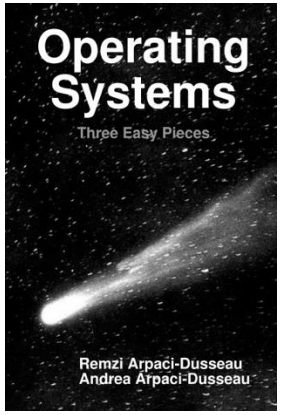
Ok, como é que isto vai funcionar?

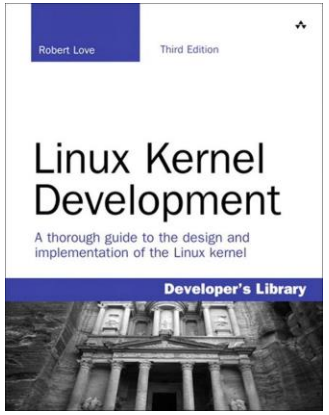
Aulas

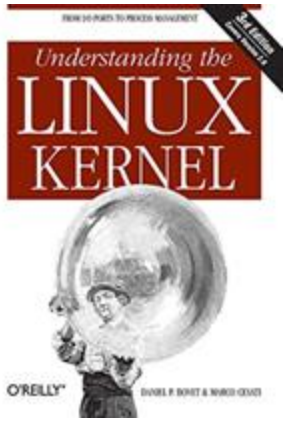
- Aulas teóricas:
  - 3 aulas de 1 h + 1 aula de 2h / semana
- Aulas laboratório:
  - 2 aulas de 1,5 h / semana
  - Já começaram esta semana
- Todas as aulas são presenciais

Bibliografia







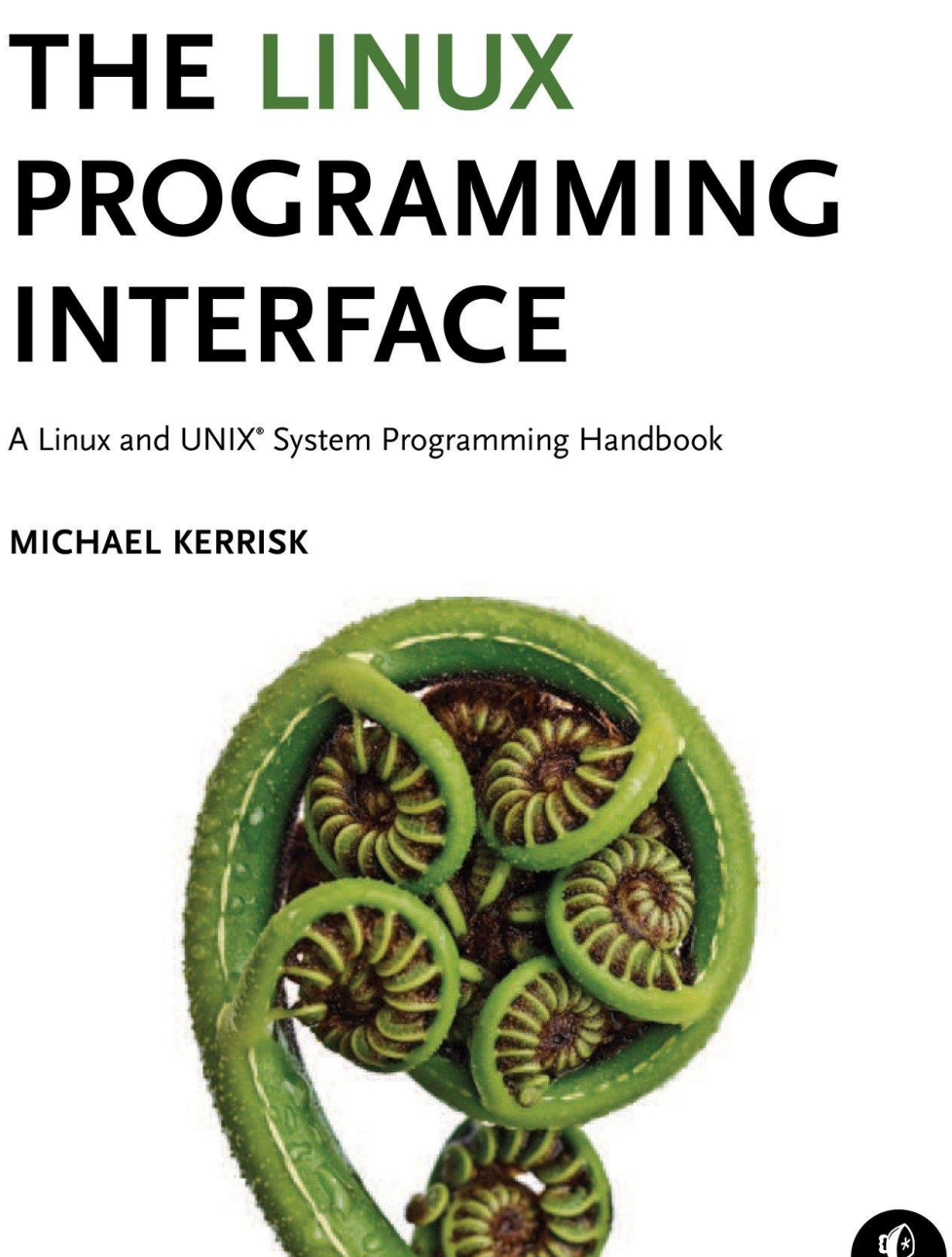


Programas de Exemplificação

O livro é uma referência para
a interface Linux

Online encontram-se todos
os programas usados no livro

http://man7.org/tlpi



Método de Avaliação (1)

- A avaliação da disciplina tem 2 componentes:
  - teórica (50%)
  - projeto (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: 8,0 valores

Método de Avaliação (3)

- Componente Projeto
  - Grupos de 2 alunos
    - Formados na 1ª 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 >=10,0v
  - Presença obrigatória na discussão final
    - inclui teste prático: extensões ao código submetido

Método de Avaliação (4)

- Avaliação do projeto:
  - Cada exercício recebe nota preliminar, meramente indicativa
  - Nota final é obtida após a discussão e teste prático
    - Avaliam a capacidade de **entender e alterar** o código do próprio projeto
    - 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

Método de Avaliação (5)

- Época Especial
  - Teórica: Exame (50%)
  - Laboratorial: projecto feito nos laboratórios ao longo do semestre (50%); ou exercício prático
- Trabalhadores-estudante
  - Avaliados de forma análoga aos restantes

Datas

- Exames
  - Época normal, recurso:
    - 24/01/2025, 15:30
    - 07/02/2025,15:30
- Projeto:
  - 1º enunciado: 12 Novembro
  - Entrega 1ª parte/2º enunciado: 12 Dez., 23h59
  - Entrega 2º parte: 9 Janeiro, 23h59
  - Teste prático e discussão final: 14 Jan. a 16 Jan.