

# Introdução aos Sistemas Operativos

Computadores e Redes de Comunicação  
Mestrado em Gestão de Informação, FEUP 06/07

**Sérgio Sobral Nunes**

mail: [sergio.nunes@fe.up.pt](mailto:sergio.nunes@fe.up.pt)

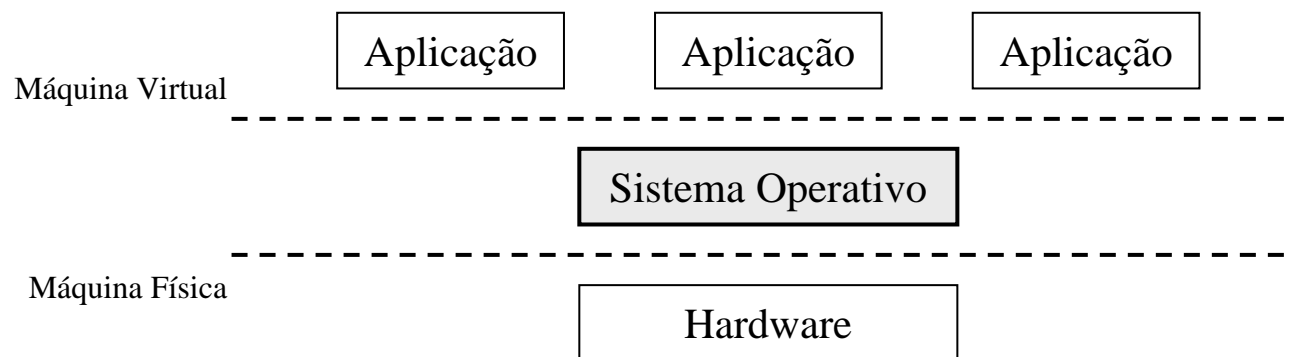
web: [www.fe.up.pt/~ssn](http://www.fe.up.pt/~ssn)

# Sumário

- Definição
- Breve história
- Camadas
- Processos
- Escalonamento
- Gestão da memória
- Memória virtual
- Sistema de ficheiros

# Definição

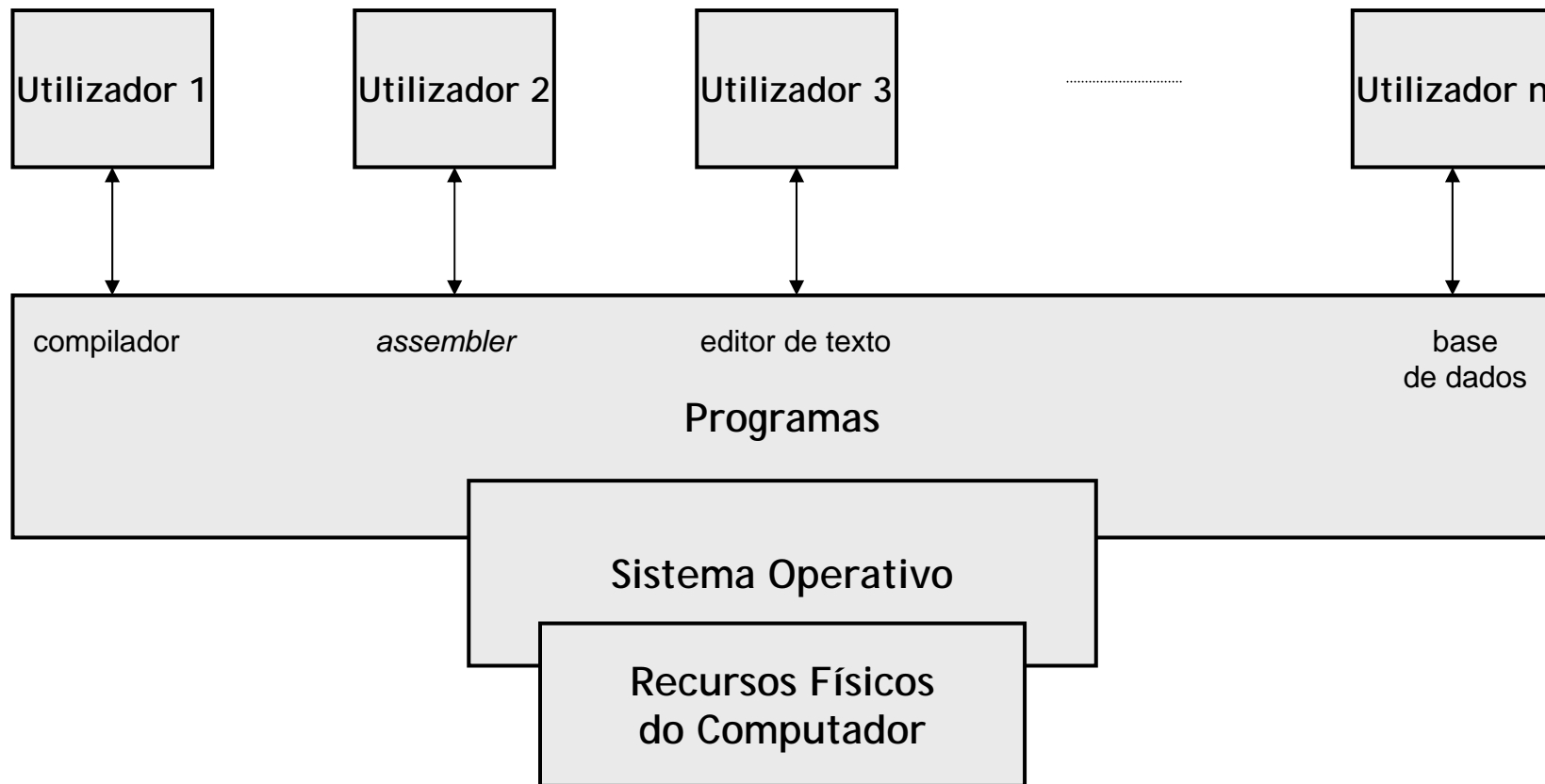
- **Um programa que funciona como intermediário entre o utilizador do computador e os recursos físicos.**
- **Objectivos:**
  - Executar programas do utilizador e tornar mais fácil a resolução de problemas. Suporte e gestão dos recursos lógicos (*software*).
  - Tornar conveniente a utilização do computador.
  - Utilizar os recursos físicos (*hardware*) do computador de forma eficiente.



# Componentes de um Sistema de Computação

1. **Recursos Físicos** – fornece os recursos computacionais base (CPU, memória, dispositivos de E/S).
2. **Sistema Operativo** – controla e coordena o uso dos recursos entre os diversas programas, dos diversos utilizadores.
3. **Programas** – definem as formas de utilização do sistema com o objectivo de resolver os problemas concretos dos utilizadores (compiladores, sistemas de bases de dados, jogos, processadores de texto).
4. **Utilizadores** (pessoas, máquinas, outros computadores).

# Componentes de um Sistema de Computação



# Alternativa ao Sistema Operativo

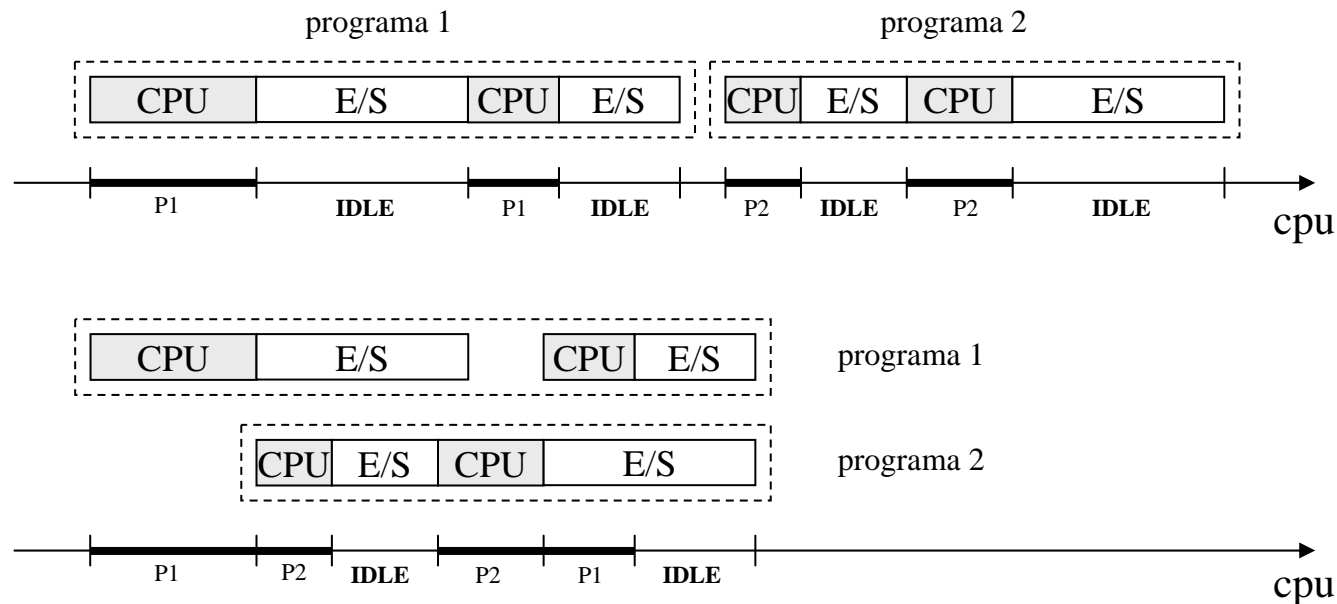
- Os programas teriam de incluir todo o código necessário para utilizar directamente a máquina física.
- Problemas:
  - Esforço de programação muito elevado.
  - Repetição de tarefas.
  - Dependente de alterações à máquina física.
  - Apenas optimizações locais, programa a programa.
  - Impossível implementar funcionalidades globais.

# Breve História

- Primeiros sistemas
  - Sem sistema operativo. Programação feita manualmente. Muito trabalho de preparação.
  - A maior parte do tempo o CPU está parado.
  - Exemplos: ENIAC, UNIVAC.
- Sistemas por lotes (*batch*)
  - Substituição do trabalho manual do operador.
  - Primeiros sistemas operativos: sequenciamento automático das operações através de instruções de controlo (LOAD, START, END).
  - CPU frequentemente parado devido às operações de E/S.
- Sistemas de multiprogramação
  - Aproveitar as esperas por tarefas de E/S.
  - As tarefas são organizadas de modo a que o CPU tenha sempre algo a executar.
  - O SO decide o escalonamento das tarefas em memória principal.
  - Exemplo: Multics.

# Multiprogramação

- Num sistema monoprocessador, apenas é executada uma *instrução* de cada vez.
- A multiprogramação permite a execução de *programas* em simultâneo.





# Breve História (cont.)

- Sistemas de partilha de tempo (*time-sharing*)
  - Ambiente interactivo.
  - Múltiplos utilizadores, cada um com um terminal.
  - É possível porque nas aplicações interactivas, a utilização do CPU é pequena.
- Sistemas pessoais
  - Sistemas dedicados exclusivamente a um utilizador.
  - Não necessita de funcionalidades tão sofisticadas (protecção, segurança).
  - Podem executar vários SO.
- Sistemas paralelos
  - Sistemas com mais do que um processador.
  - Partilha da memória, relógio, periféricos, canais de comunicação.
  - Vantagens: maior débito, económico, fiabilidade.
  - Problemas: custos de coordenação, programação específica.

# Breve História (cont.)

- **Sistemas distribuídos**
  - Computação é distribuída por vários processadores autónomos.
  - CPUs independentes (memória, barramento) – *desacoplamento*.
  - Requer infraestrutura de rede (LAN, WAN).
  - Vantagens: partilha de recursos, partilha de carga, fiabilidade.
- **Sistemas de tempo real**
  - Restrições temporais bem definidas.
  - Rígidos (garante): controlo de experiências científicas, sistemas de imagem médica, controlo industrial.
  - Flexíveis (prioridades): aplicações multimédia, realidade virtual.
- **Sistemas para “dispositivos móveis”**
  - PDAs, telemóveis.
  - Questões: memória limitada, processadores lentos, ecrãs pequenos.
  - Exemplo: Symbian OS.

# Arquitectura

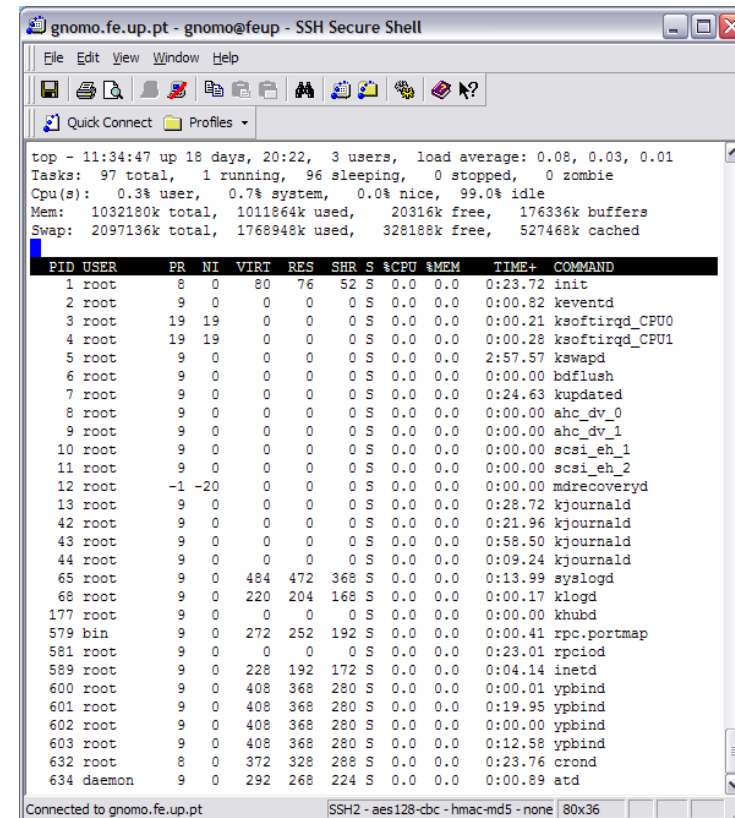
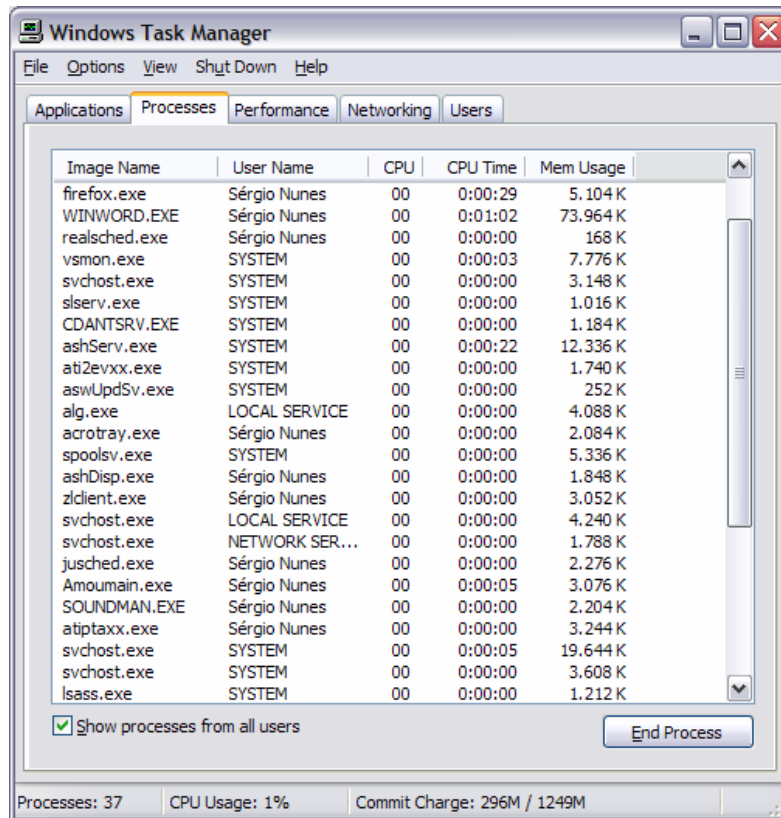
- As principais funções de um sistema operativo estão estruturadas em camadas.
- O uso de arquitecturas modulares facilita o desenvolvimento de funcionalidades por camadas.

Nível	Nome	Funções
5	Interpretador de comandos	Interface entre o SO e o utilizador.
4	Sistema de ficheiros	Chamadas de alto nível para manipulação de ficheiros, impressoras, terminais.
3	Gestão de memória	Gestão de memória, memória virtual, <i>swap</i> , protecção.
2	E/S básica	Funcionalidades de baixo nível para o nível seguinte.
1	Núcleo (kernel)	Escalonamento, mudança de contexto, interrupções, sincronização básica.

# Processos

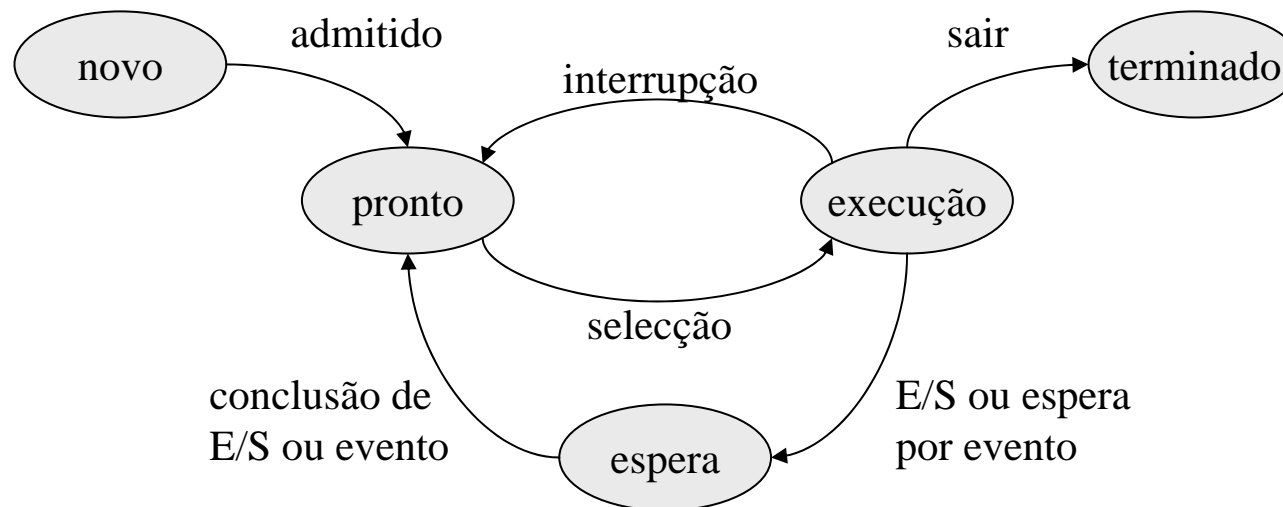
- Um processo é um programa em execução. Um processo necessita de certos recursos, incluindo tempo de CPU, memória, ficheiros e dispositivos de E/S para levar a cabo a sua tarefa.
- O mesmo programa pode estar a ser executado por diferentes processos (múltiplos utilizadores).
- O sistema operativo é responsável por:
  - Criação e eliminação de processos.
  - Suspensão e (re-)arranque de processos.
  - Fornecer mecanismos para a sincronização e comunicação entre processos.

# Processos



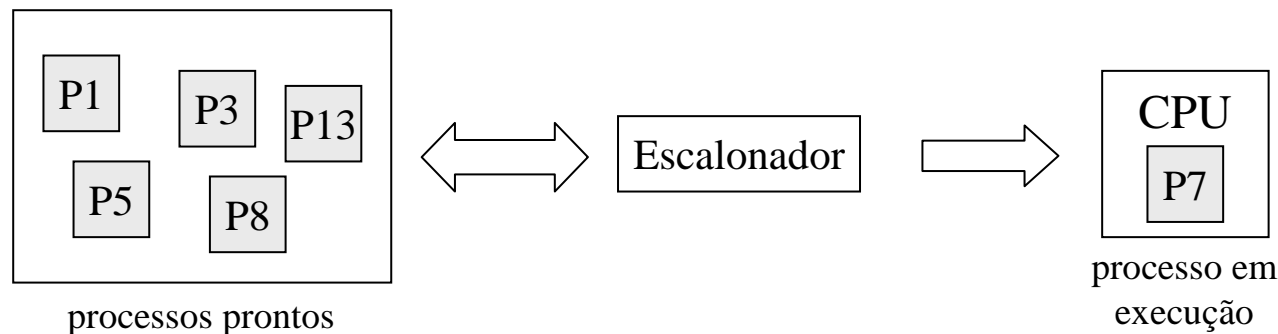
# Processos

- Durante a execução, um processo muda de estado.



# Escalonamento

- Associado ao conceito de multiprogramação.
- Sempre que o CPU está livre, cabe ao escalonador seleccionar um dos processos prontos para execução.
- Dois tipos:
  - Preemptivo: pode retirar o processo sem que tenha terminado.
  - Não-preemptivo: espera que o processo liberte o CPU.
- Exemplos de critérios: *first-come first-served* (FCFS), *shortest-job-first* (SJF), prioridade, *round-robin*.



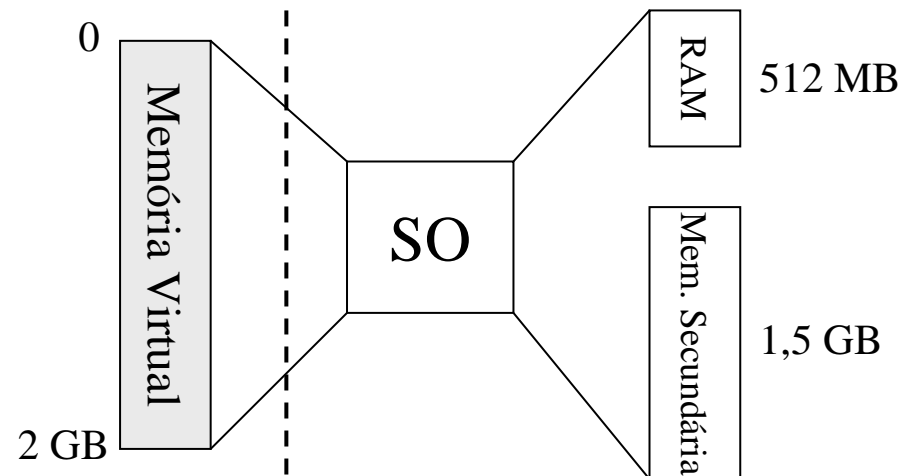
# Gestão de Memória

- Memória corresponde a uma grande colecção de bytes, cada um com endereço próprio.
- Um programa deve ser carregado para a memória e colocado dentro de um processo para que possa ser executado.
- O sistema operativo é responsável por:
  - Manter informação sobre que partes da memória estão a ser usadas, e por quem.
  - Decidir a que processos atribuir espaço livre de memória.
  - Atribuir e libertar memória de acordo com as necessidades.



# Memória Virtual

- Espaço de endereçamento que engloba a memória primária e secundária.
- Tira partido do facto do espaço disponível na memória secundária ser muito superior.
- Permite a execução de programas maiores do que a memória principal.
- Implementação complexa.



# Sistema de Ficheiros

- Um sistema de ficheiros é constituído por duas partes:
  - Um conjunto de ficheiros (dados).
  - Uma estrutura de directórios (organização).
- O sistema operativo é responsável por:
  - Criação e eliminação de ficheiros.
  - Criação e eliminação de directórios.
  - Suporte das funções base para a manipulação de ficheiros e directórios.
  - Mapeamento dos ficheiros (lógico) no suporte de armazenamento secundário (físico).
  - Criação de cópias de segurança dos ficheiros em suporte estável, não volátil.
  - Gestão e manutenção das permissões de acesso aos ficheiros.

# Sistema de Ficheiros

- Um ficheiro é uma colecção de informação relacionada, identificada por um nome. Normalmente, os ficheiros representam programas ou dados.
- Propriedades dos ficheiros:
  - Nome
  - Tipo
  - Local
  - Tamanho
  - Segurança
  - Data
  - Dono
- Operações sobre ficheiros:
  - Criação
  - Escrita
  - Leitura
  - Posicionamento
  - Eliminação

# Sistema de Ficheiros

- A estrutura de directórios permite gerir e organizar o conjunto de ficheiros existentes num *volume*.
- Operações desempenhadas num directório:
  - pesquisa de ficheiros, criação de ficheiros, eliminação de ficheiros, listagem de conteúdos, renomear ficheiros.
- São possíveis muitas formas de organização. A mais comum é em árvore com um número arbitrário de níveis.

# Sistema de Ficheiros

