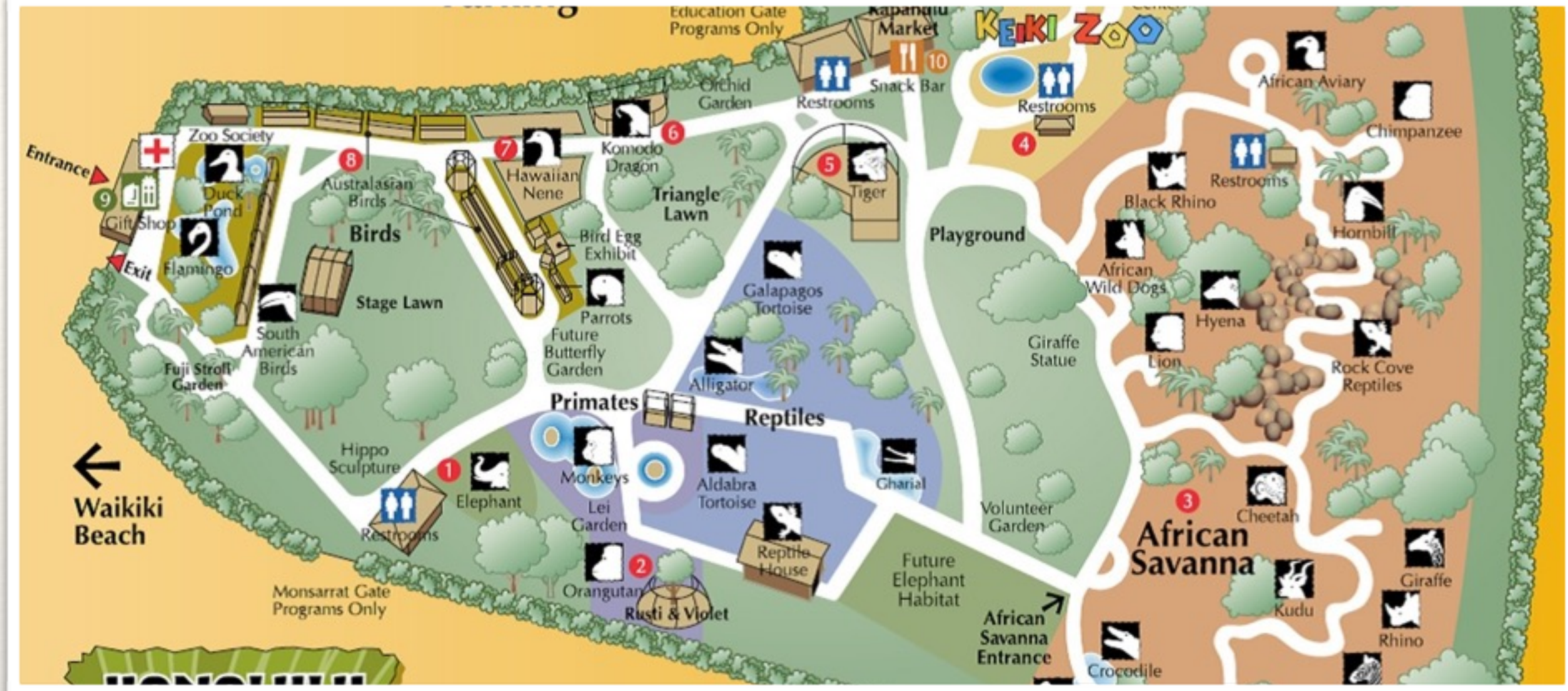
Prof. Ricardo Inácio Álvares e Silva

Principais Conceitos

Sistemas Operacionais

Escopo da aula

- Tipos de Sistemas Operacionais
- Principais Conceitos
- Estruturas de Sistemas Operacionais



Diversos Sistemas Operacionais

Tipos de Sistemas Operacionais

O zoológico dos SO…

Mainframes

- Especializados em E/S
  - Dezenas de milhares de discos
- Tempo compartilhado
- Lotes de tarefas (jobs)
- Milhões de pequenas transações
  - Bancos, reservas de ingressos, web servers, etc
- Exemplos atuais: System Z, s390, AIX, Solaris, HP-UX, BSD e Linux

Servidores

- Provém serviços a uma rede de computadores
  - Impressão
  - Sistema de arquivos compartilhados
  - Web server
  - Repositório
  - Controle de versão
- Exemplos atuais: FreeBSD, Linux, Windows Server

Multiprocessadores e Pessoais

- Feitos para computadores paralelos
  - múltiplos processadores
- Cuidados especiais de consistência e comunicação
- Aplicativos precisam se adaptar para tirar vantagem das capacidades do sistema
- Sistemas Pessoais
  - atualmente é o mesmo de multiprocessadores

Plataformas Móveis

- Tablets, smartphones, etc.
- Atualmente similares aos Sistemas Pessoais
- iOS, Android, Symbian, Windows Phone, etc

Embarcados

- Sistemas que dão suporte ao funcionamento de diversos aparelhos, desde domésticos a industriais
- TVs, DVDs, MP3 Players, carros, celulares, etc
- **Não aceitam aplicativos externos**
- Normalmente são monoprogramados, sem modo núcleo

Redes de Sensores

- Redes de pequenos computadores
  - Vigiam um perímetro
  - Verificam incêndios
  - Clima
- Tolerantes a falhas
- Funções simples

Sistemas de Tempo Real

- Sistemas para fábricas
- Processos que precisam ser executados em tempo exato
- Garantem que um programa será iniciado e terminado em tempo exato, não aceita falhas
- Verificação de linha de produção

Sistemas de Smart Cards

- Severamente limitado por
  - energia
  - capacidade computacional
- Funciona por um curtíssimo período de tempo

Principais Conceitos

Conceitos de Sistemas Operacionais

- A maioria dos sistemas operacionais provê conceitos e abstrações similares:
  - Processos
  - Espaços de Endereçamento
  - Arquivos
  - Entrada e Saída (E/S)
  - Proteção

Processos

- É uma instância de um programa em execução
- Exemplo:

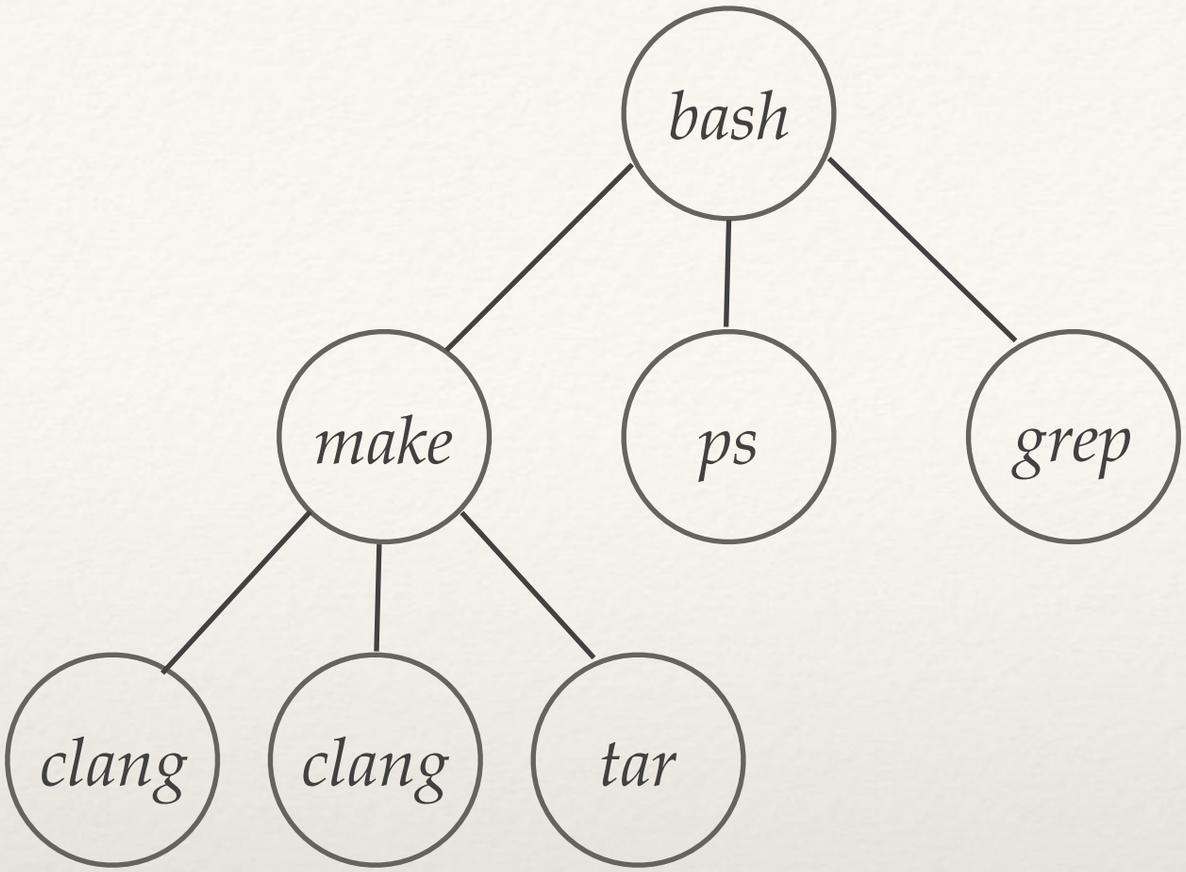
  "Em um ambiente multiprogramado, um usuário inicia um editor de vídeos. Ele ordena instrui a utilização de um efeito especial cuja aplicação levará pelo menos uma hora, e então abre o navegador Web. Enquanto isso, um processo em segundo plano que desperta periodicamente para verificar chega de emails é ativado. Dessa forma, tem-se três processos ativos. Periodicamente, o sistema operacional decide parar um deles para continuar outro."

- Quando um processo é suspenso dessa forma, deve ser reiniciado depois exatamente no mesmo estado em que se encontrava

- Cada processo tem associado um **espaço de endereçamento**, locais de memória onde ele pode ler e escrever
- Possuem também outros **recursos** associados:
  - Registradores da CPU
  - Lista de arquivos abertos
  - Alarmes programados
  - Lista de processos relacionados
  - Outras informações necessárias
- Um processo é fundamentalmente um **contentor** que guarda as informações necessárias a execução de um programa
- Toda a informação sobre um processo deve ser salva quando ele for suspenso, normalmente na **tabela de processos**

Chamadas ao Sistema p/ Processos

- As principais chamadas ao sistema para gerenciamento de processos são as que controlam sua criação e finalização
- Exemplo:
  - A linha de comando, ou terminal, é um processo
    - Quando o usuário utiliza um comando para compilar algum código, o processo do terminal cria um novo processo para o compilador
    - Quando o compilador terminar sua tarefa, ele invoca o fim de seu próprio processo
- Quando um processo pode criar um ou mais **processos filhos**, ele produz uma árvore de processos
- É comum vários processos cooperando para uma tarefa, e precisam trocar informações. Essa operação é conhecida por **comunicação entre processos**



Uma possível árvore de processos no UNIX

Faça o seguinte comando no terminal:

```
$ ps aux | grep -i python3
```

Comunicação entre processos através de ***pipe***

- Um processo pode ser acionado através de **sinais** do sistema operacional
- Um processo pode programar um sinal de alarme, que passado um certo tempo, o aciona
- Outras chamadas ao sistema:
  - Requisitar mais memória, ou liberar
  - Aguardar o término de um processo filho
  - Trocar o programa em execução

Pipes

- Uma maneira de dois processos se comunicarem é através de *pipes*
- Devem ser definidos no lançamento dos processos
- Cada processo funciona normalmente, como se estivesse recebendo dados de um dispositivo E/S (ex. teclado) e escrevendo em outro (imprimindo na tela)
- Ao invés de imprimir na tela, o dado é enviado pelo pipe para o outro processo
  - Ex: `cat o_hobbit.txt | less`

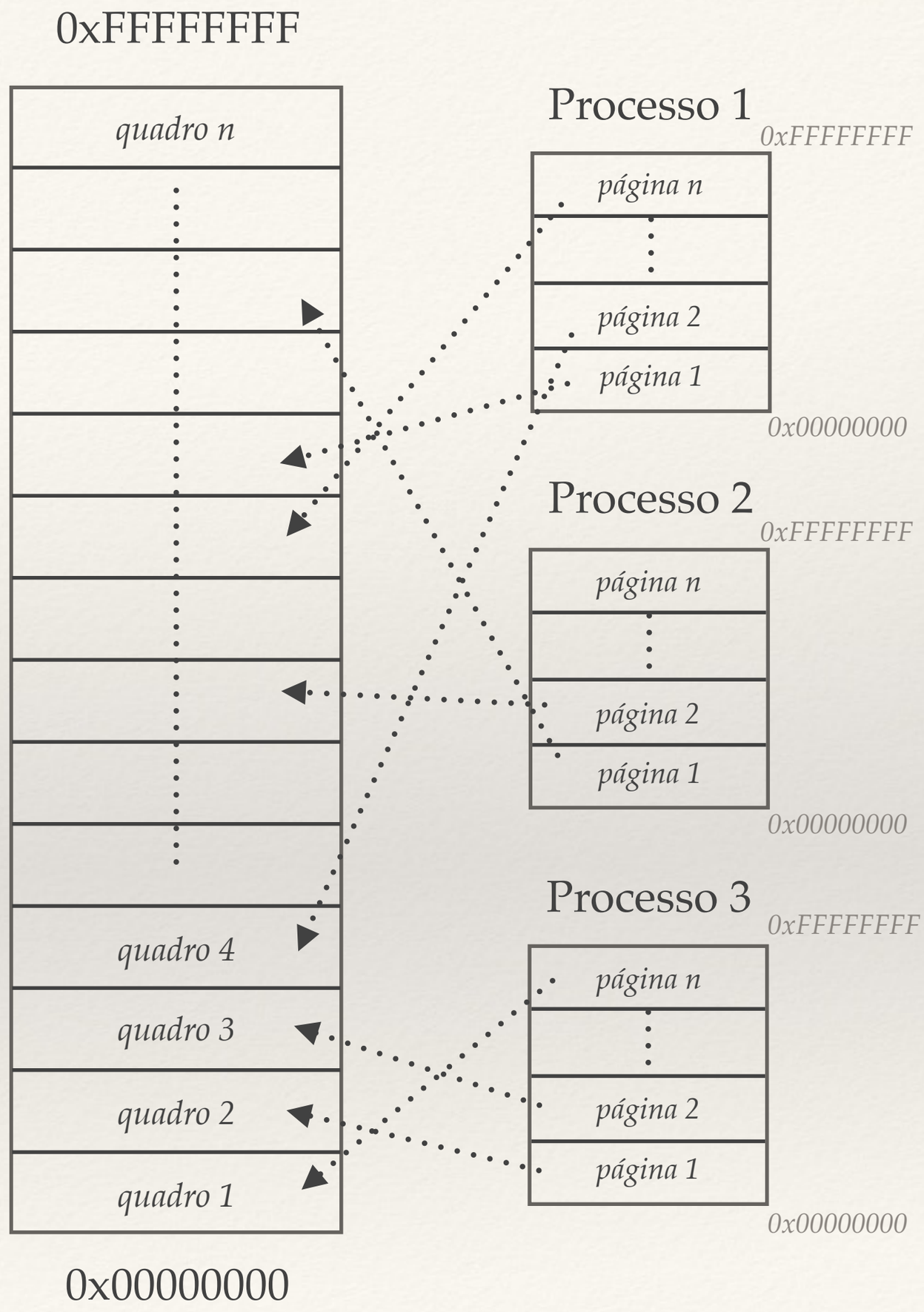
Identificação de Processos

- Todo usuário possui um **número de identificação, UID**
- Todo processo iniciado por um usuário herda o seu UID, assim como os seus processos filhos
- Pelo menos um UID, o do administrador, tem poderes especiais e pode violar várias regras de proteção
- Alguns sistemas possuem organização de **grupos de usuários**, identificados por um número **GID**, com funcionamento similar ao de UID.

Espaços de Endereçamento

- Todo computador possui uma memória principal, para onde carrega os programas a serem executados:
  - sistemas simples só carregam um programa por vez
    - Exemplos: DOS e CP/M
  - sistemas complexo carregam múltiplos programas
- Mecanismo de proteção em hardware, mas controlado pelo sistema operacional
  - O que fazer quando um processo tentar acessar a memória de outro?

- Muitos sistemas possuem endereçamento de 32 ou 64 bits, mas possuem menos memória física instalada do que isso
- O que acontece em um sistema 32 bits, com 1 GB de RAM se um processo precisar de 2 GB de RAM?
    A. Tem o pedido negado
    B. Separa o processo em páginas e utiliza quadros da memória sob demanda.
- Esta opção B é conhecida como **memória virtual**
- A imagem ao lado ilustra o funcionamento

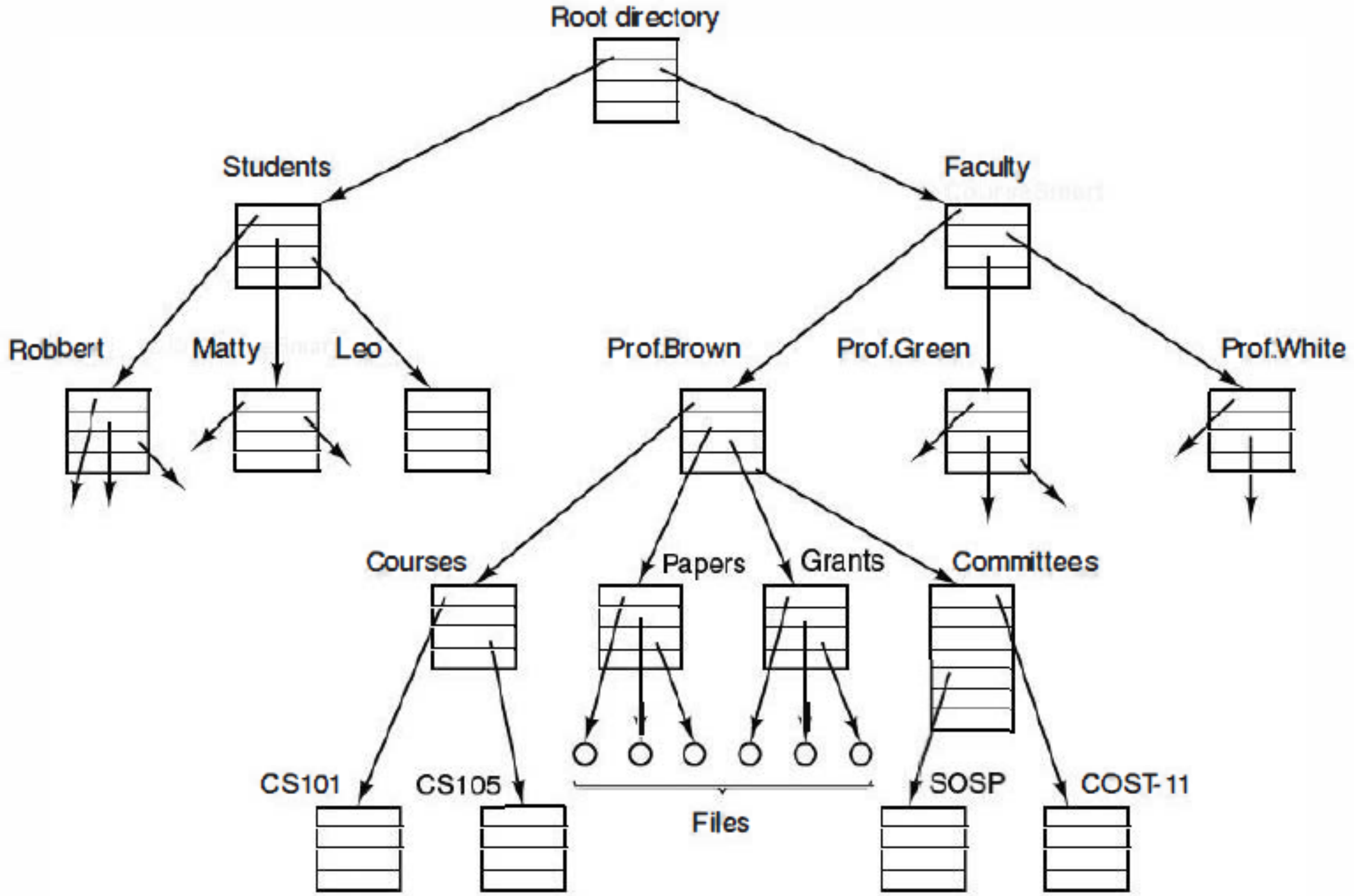


Arquivos

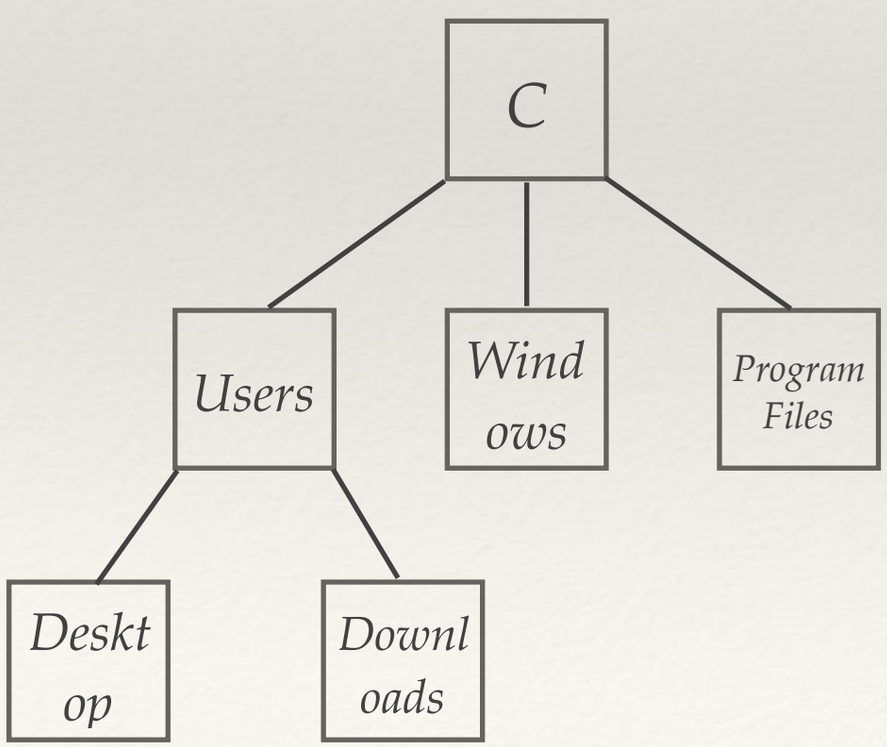
- Dentre as principais funções de S.O. está a abstração de utilização de discos e outros dispositivos de E/S
- Chamadas ao sistema intermediam operações com arquivos
  - Criação
  - Remoção
  - Leitura
  - Escrita
- Antes de serem utilizados, precisam ser abertos, também através de chamadas ao sistema
- Após uso, devem ser fechados
- **Descritor de arquivo** controla estado e operações utilizadas

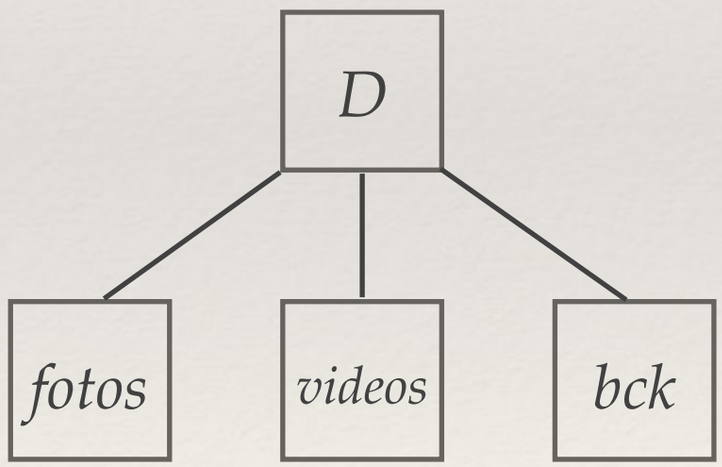
Organização de Arquivos

- Através de **diretórios** e **hierarquia**:
- Chamadas de sistema também controlam diretórios
  - criar, mover, apagar
- Todo arquivo em um diretório dentro de uma hierarquia pode ser acessado por seu **nome de caminho**, a partir do **diretório raiz**
  - Ex: `/Usuários/Ricardo/Documentos/Unifil/SO/Prova01.pdf`
- Todo processo possui um **diretório de trabalho**
  - Ex: Paint têm como diretório de trabalho `/Usuários/Ricardo/Imagens/`
  - Se ele quiser abrir o arquivo `/Usuários/Ricardo/Imagens/Praia/foto1.png`, só precisa especificar: `Praia/foto1.png`
- Diretório de trabalho pode ser mudado com chamadas ao sistema

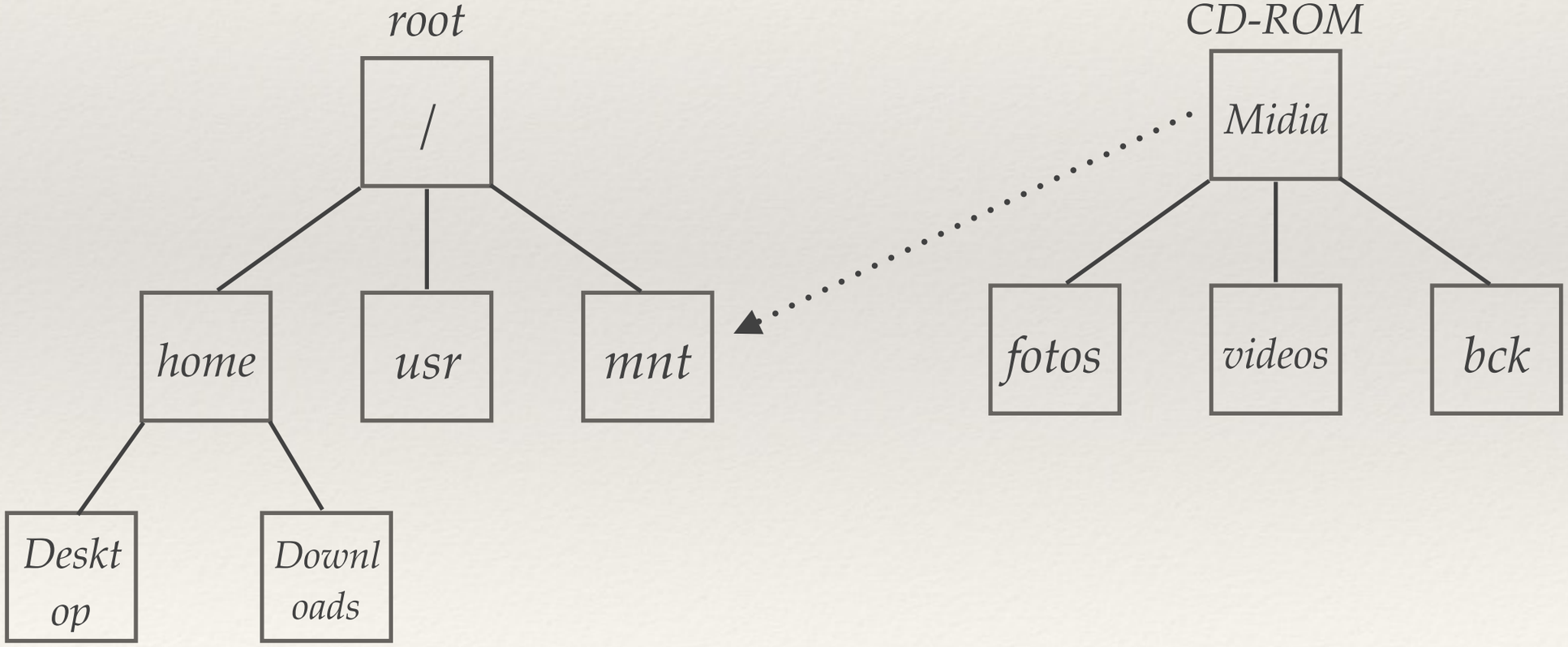

Root directory
Students
Faculty
Robbert
Matty
Leo
Prof.Brown
Prof.Green
Prof.White
Courses
Papers
Grants
Committees
CS101
CS105
Files
SOSP
COST-11

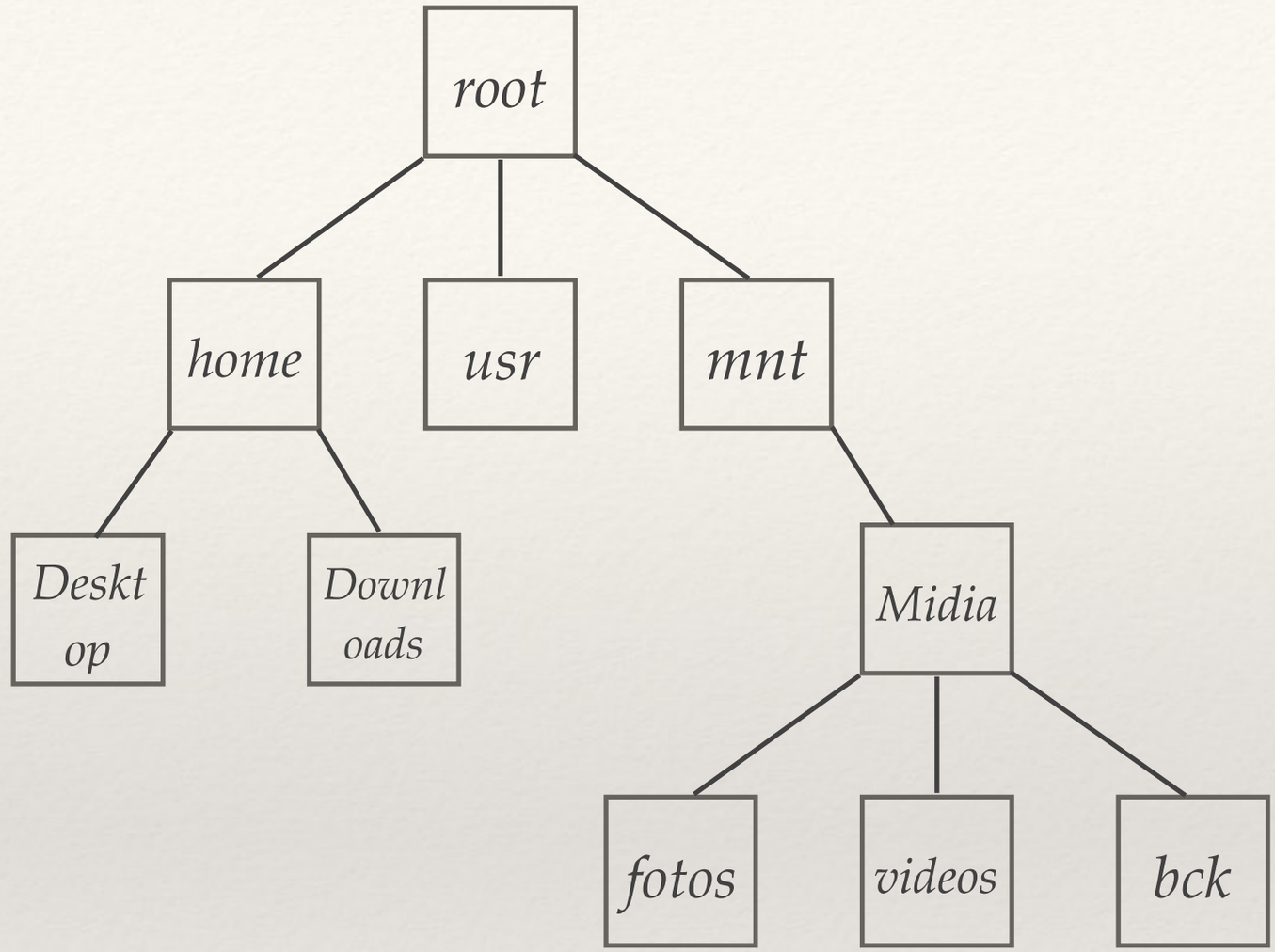
- Sistemas de arquivos **montados**
  - CD-ROM
  - DVD-ROM
  - USB
  - Imagem de outro sistema de arquivos
- Alguns sistemas (ex, Windows 95 e Symbian) criam uma árvore de diretórios para cada dispositivo, o que pode gerar dependência de organização de dispositivos





- Outros sistemas (ex, UNIX) **montam** árvores distintas à árvore principal (*root*)
  - Enquanto não estiverem montadas, as árvores distintas não são acessíveis




root
home
usr
mnt
Deskt
op
Downl
oads
Midia
fotos
videos
bck

Arquivos especiais

- Conceito utilizado por UNIXes, um **arquivo especial** é uma maneira de um processo se comunicar com um dispositivo, mesmo que ele não seja um arquivo real.
- **Arquivos especiais de bloco**: modelam dispositivos que possuem acesso aleatório, como os discos
  - Qual a diferença de acesso ao disco por um arquivo comum ou por um arquivo especial?
- **Arquivos especiais de caractere**: modelam dispositivos que se comunicam por sequência de caracteres, comandos e dados
  - Impressoras, modems, adaptadores de áudio, etc

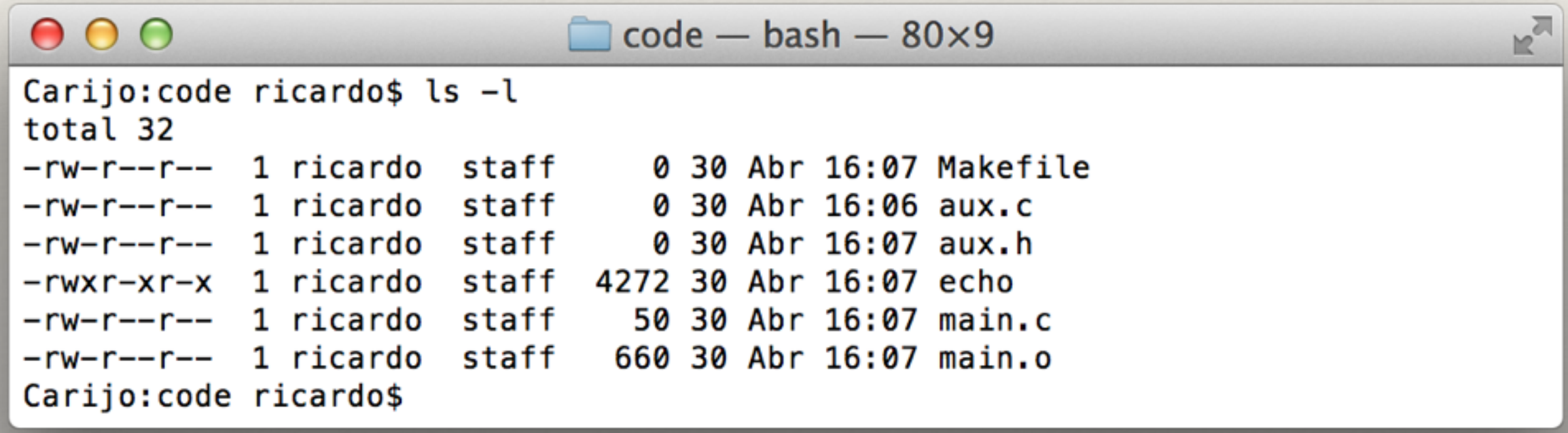
Entrada e Saída (E/S)

- Todo computador possui dispositivos de E/S
- Cabe ao S.O. gerenciá-los, por isso todos possuem um subsistema para essa finalidade
- Alguns dispositivos funcionam igualmente bem sob um mesmo conjunto de chamadas de sistema, ou modelados como arquivos
- Outros dispositivos devem receber tratamento especial, através de **drivers de dispositivo**

Proteção

- A proteção aos dados dos usuários e do próprio sistema é tarefa do S.O.
- Arquivos contém conjunto de permissões de acesso
- No UNIX cada arquivo possui um **descritor de permissões 9-bit**
  - São 3 grupos de 3 bits: (000) (000) (000)
  - O grupo da esquerda define permissões do proprietário do arquivo
  - O grupo do meio define permissões do grupo proprietário do arquivo
  - O grupo da direita define permissões para todos os outros usuários

- Os 3 bits de cada grupo possuem o seguinte significado:
  - Bit 3: permissão de leitura, indicado por ‘r’
  - Bit 2: permissão de escrita, indicado por ‘w’
  - Bit 1: permissão de execução, indicado por ‘x’


code — bash — 80×9
Carijo:code ricardo$ ls -l
total 32
-rw-r--r-- 1 ricardo staff   0 30 Abr 16:07 Makefile
-rw-r--r-- 1 ricardo staff   0 30 Abr 16:06 aux.c
-rw-r--r-- 1 ricardo staff   0 30 Abr 16:07 aux.h
-rwxr-xr-x 1 ricardo staff 4272 30 Abr 16:07 echo
-rw-r--r-- 1 ricardo staff   50 30 Abr 16:07 main.c
-rw-r--r-- 1 ricardo staff   660 30 Abr 16:07 main.o
Carijo:code ricardo$