Big Data & Data Science

Infraestrutura Computacional
Parte 1: GNU/Linux e Shell

Introdução ao GNU/Linux

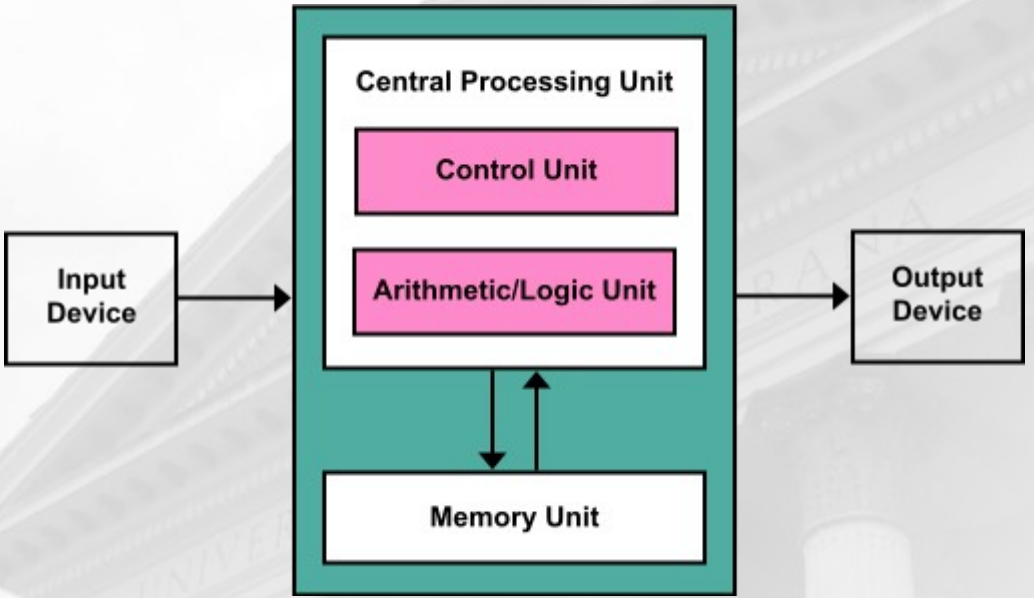
DSBD

Apresentação

Prof. Lucas Ferrari de Oliveira

- lferrari@inf.ufpr.br
- web.inf.ufpr.br/lferrari
- Áreas de Pesquisa
  - Processamento de Imagens
  - Processamento de Imagens Médicas
  - Reconhecimento de Padrões

Arquitetura de von Neumann



UNIX

No princípio (1970) era o UNIX...

- Sistema Operacional criado no AT&T Bell Labs
- Introduziu e popularizou conceitos poderosos
  - sistema de arquivos, shell, processos, usuários
- Por volta de 1990
  - Patentes e copyright isolaram UNIX em nichos
  - Não era compatível com PCs (x86)
  - UNIX foi padronizado (POSIX)

GNU

… e o UNIX se fez GNU, e habitou entre nós

- **G**NU's **N**ot **U**nix: conjunto de programas FOSS compatíveis com POSIX e funcionalidade similar ao UNIX
  - Shell (interpretador de comandos)
  - Utilidades básicas UNIX: `cp`, `mv`, `cat`, `ls`, `awk`, `sed`, `grep`, `less`, `man`, `kill`, `ps`, `chmod`
  - Editor de textos (Emacs, vi)
  - Interface Gráfica (GNOME)

Linux

Todo SO precisa de um *kernel*, que controle o hardware

- Linux foi criado em 1992 por Linus Torvalds para x86
- Compatível com UNIX: mesma API de chamadas de sistema, design e arquitetura semelhantes
- Programas GNU podiam ser compilados e rodar em x86
- Distribuições = kernel + software
  - GNU/Linux
  - Debian, Slackware, SUSE, RedHat, Fedora, Ubuntu, CentOS, Mint

Por que GNU/Linux?

Boa implementação de excelentes ideias UNIX

Grande comunidade de Software Livre

- adicionando funcionalidades, suporte a hardware, correção de *bugs*, testando

Licensa GPL permite uso mas exige distribuição do código fonte

Alta performance, escalabilidade, suporte a grande quantidade de dispositivos

Por que GNU/Linux?

Computadores pessoais (Desktop)

- Escolha uma distribuição e experimente
  - Geração de pendrive para carga do SO
  - Instalação concomitante com outro SO

Smartfones (Android, Tizen)

Dispositivos Embarcados

- Roteadores, GPS, Raspberry Pi

Servidores WEB

Por que GNU/Linux

Supercomputadores

- TOP 500 (100% desde nov/2017)

Space X Falcon 9

International Space Station

Command Line Heroes

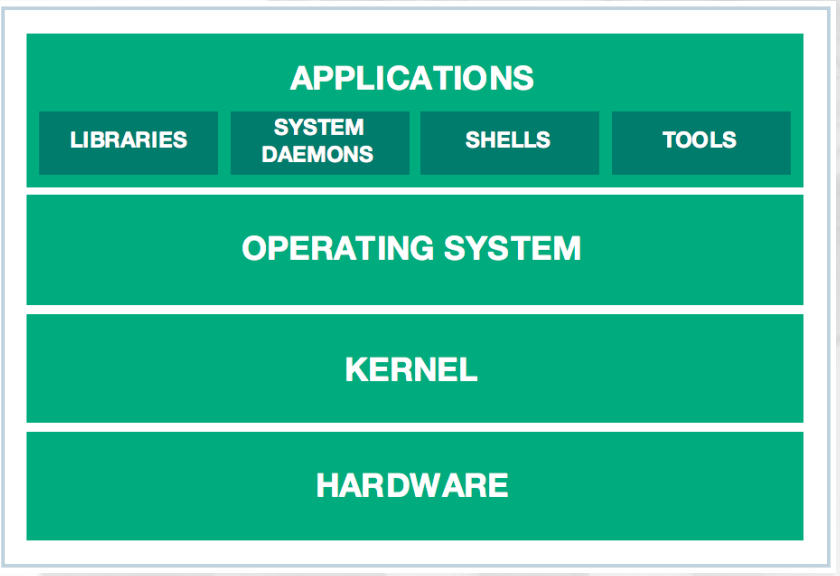
Sistema Operacional GNU/Linux

Características do GNU/Linux

- **Portável:** diferentes tipos de *hardware*
- ***Open Source*:** www.gnu.org (*copyleft*)
- **Multi usuário:** acesso simultâneo
- **Multi processos:** diversos programas simultaneamente
- **Sistema de Arquivos Hierárquico**
- **Shell:** programa interpretador de comandos
- **Segurança:** autenticação de usuários, criptografia, controle de acesso

Sistema Operacional

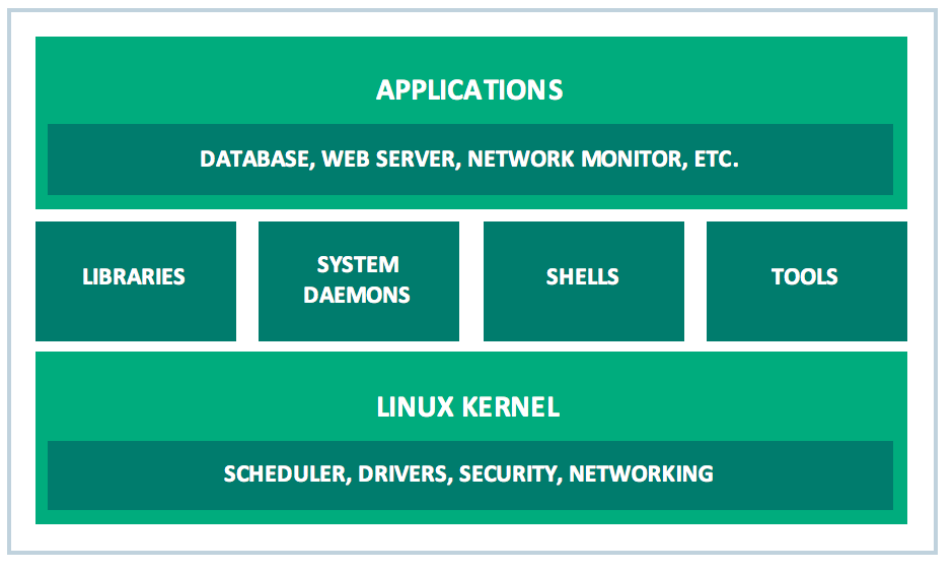
O Sistema Operacional é um software que controla o hardware e faz a interface deste com as aplicações



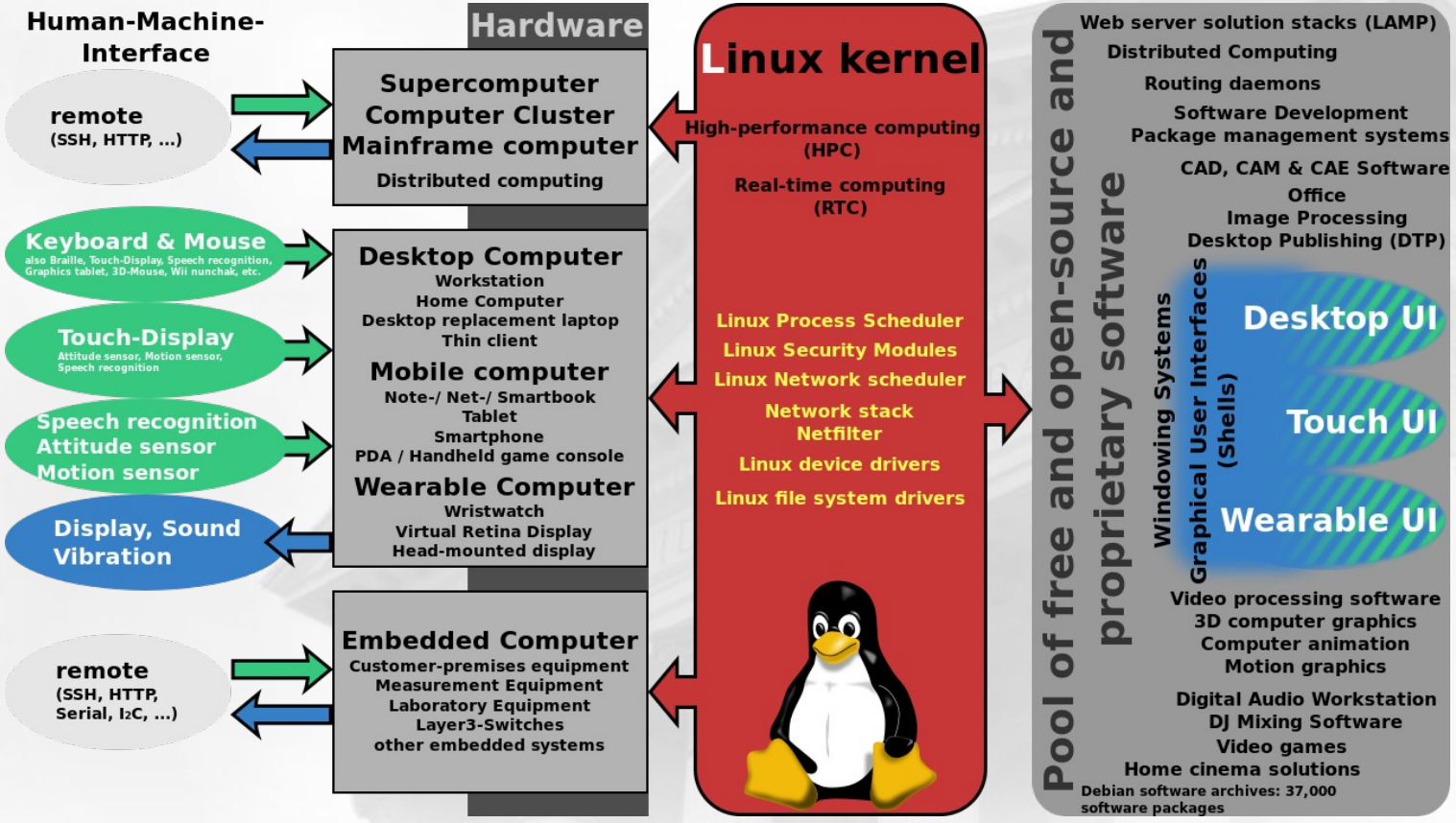
Sistema Operacional GNU/Linux

É um gerenciador de recursos composto pelo Kernel e um conjunto de aplicações básicas

- Serviços e *daemon*
- Programas utilitários (shell, editor, compilador)
- Biblitoeca C (libc)



Sistema Operacional GNU/Linux



Kernel Linux

O kernel é uma parte do SO:

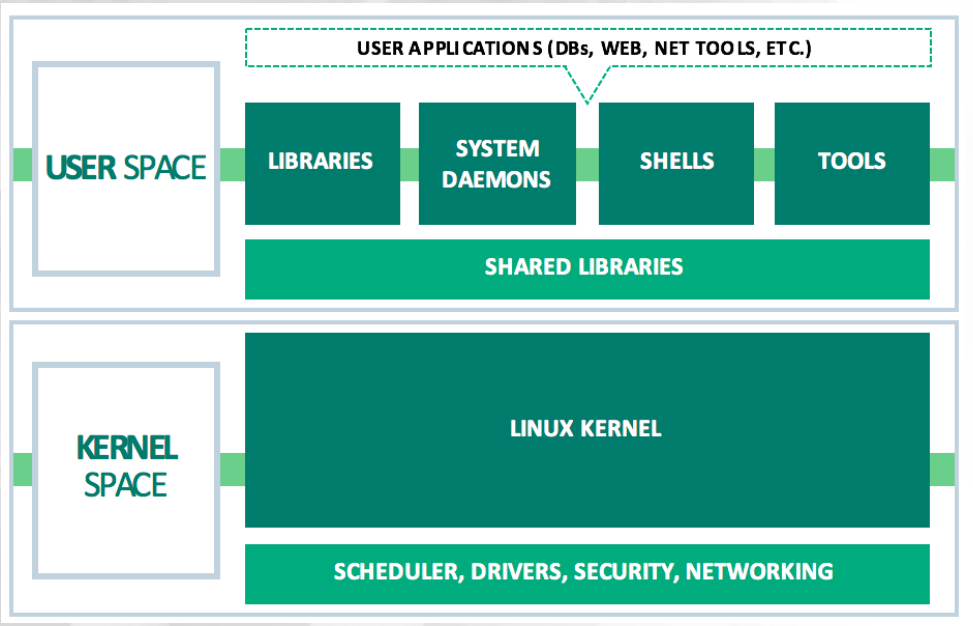
- Controla a CPU, memória e outros dispositivos
- Acessa dados em dispositivos de armazenamento
- Escalona processos
- Roda aplicações, isolando-as umas das outras
- Disponibiliza uma API (*system calls*) para atividades restritas

User × Kernel space

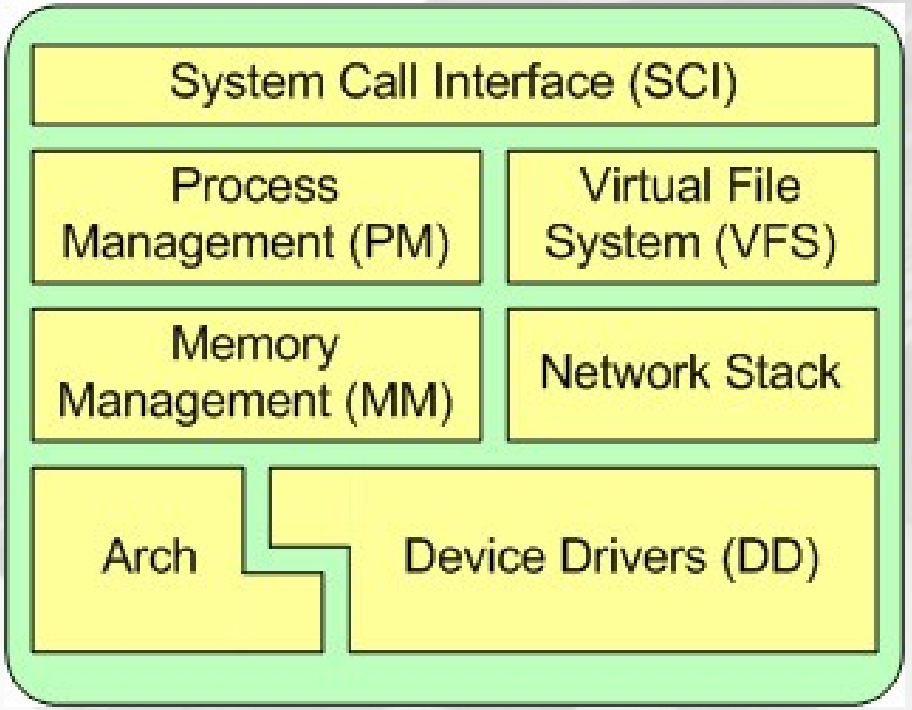
O GNU/Linux executa seu kernel (Linux) em uma região de memória restrita e protegida (*kernel space*)

Programas do SO e dos usuários rodam em outra região de memória (*user space*)

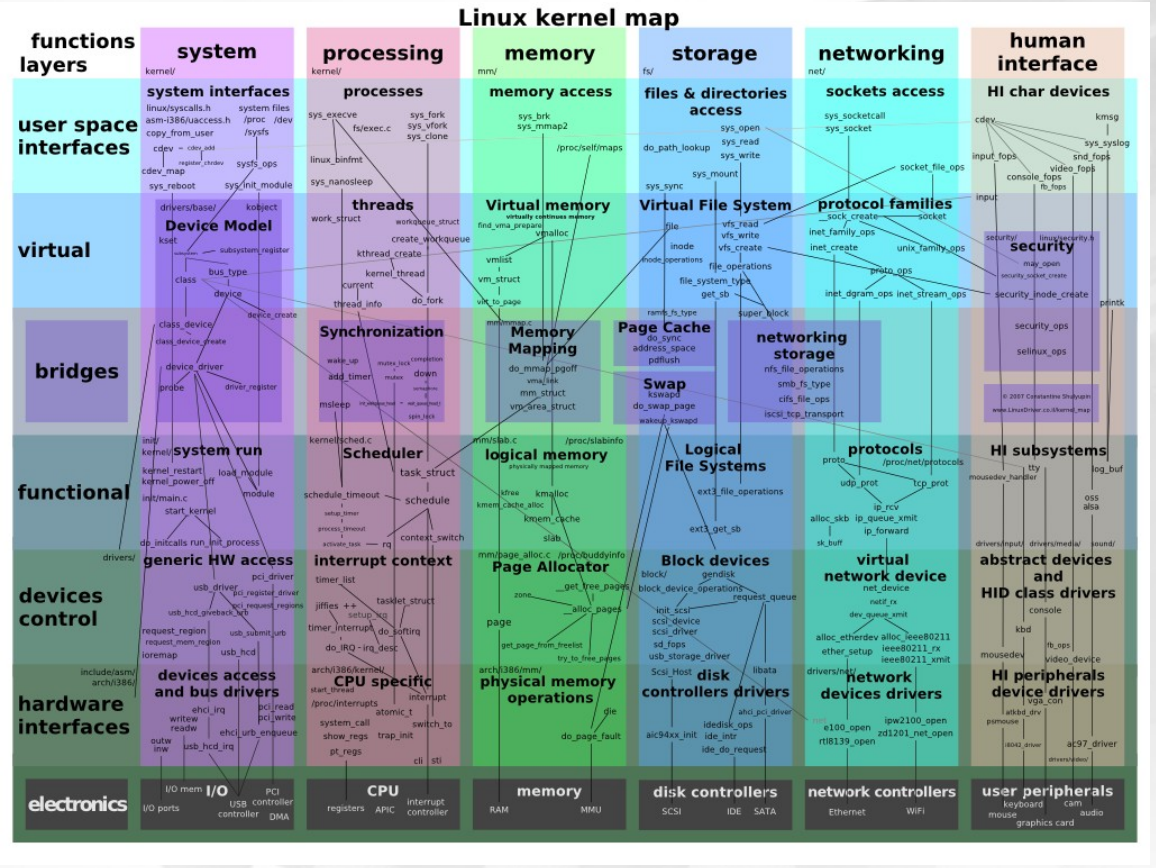
- Spectre & Meltdown



Componentes do Linux



Componentes do Linux



Componentes do Linux

Interface de *system call* (SCI)

- Funções que podem ser invocadas em *user space* para serem executadas em *kernel space*

Gerenciamento de Processos (PM)

- Executa processos ou *threads*, que são a virtualização do processador e memória
- Provê API para criação, destruição e comunicação interprocessos
- Escalona processos compartilhando o mesmo hardware

Componentes do Linux

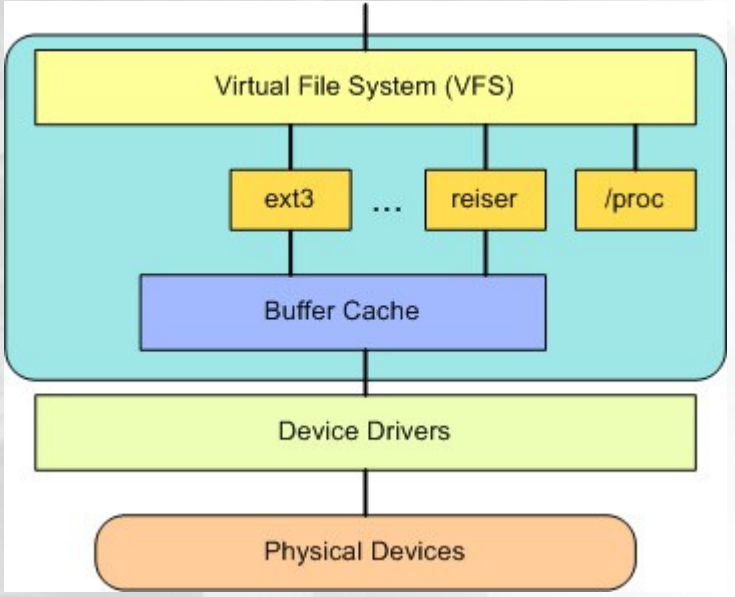
Gerenciamento de Memória (MM)

- Divide a memória em blocos e gerencia sua alocação
- Permite crescimento e redução dinâmicos da memória ocupada
- Separa memória de cada processo e usuário
- Provê memória adicional através *swap*

Componentes do Linux

Sistema de Arquivos Virtual (VFS)

- Provê uma interface abstrata comum para sistemas de arquivos (*open, close, read, write*)
  - O sistema tem um diretório raiz: `/`
- Gerencia *buffer caches* para acelerar acesso ao sistema de arquivos
- Interface para acesso ao kernel em `/proc`



Componentes do Linux

Camada de Rede

- Implementa protocolos de rede (TCP, IP, Infiniband)
- Provê uma interface chamada *socket*, que é a maneira de comunicação ponto a ponto em Linux

Drivers de dispositivos

- Software específico para acesso aos diferentes dispositivos

Código dependente de Arquitetura

Perguntas?

Computação de Alto Desempenho

High Performance Computing

HPC refere-se à prática de agregar poder computacional (diversos processadores) de forma a obter uma performance muito maior do que poderia ser obtida com um computador individual, a fim de resolver problemas de grande escala.

Escalas de grandeza

Prefix	Symbol	1000^m	10^n	Decimal	Short scale	Long scale	Since[n 1]
yotta	Y	1000^8	10^{24}	1 000 000 000 000 000 000 000 000	Septillion	Quadrillion	1991
zetta	Z	1000^7	10^{21}	1 000 000 000 000 000 000 000	Sextillion	Trilliard	1991
exa	E	1000^6	10^{18}	1 000 000 000 000 000 000	Quintillion	Trillion	1975
peta	P	1000^5	10^{15}	1 000 000 000 000 000	Quadrillion	Billiard	1975
tera	T	1000^4	10^{12}	1 000 000 000 000	Trillion	Billion	1960
giga	G	1000^3	10^9	1 000 000 000	Billion	Milliard	1960
mega	M	1000^2	10^6	1 000 000	Million		1960
kilo	k	1000^1	10^3	1 000	Thousand		1795
hecto	h	$1000^{2/3}$	10^2	100	Hundred		1795
deca	da	$1000^{1/3}$	10^1	10	Ten		1795
		1000^0	10^0	1	One		–
deci	d	$1000^{-1/3}$	10^{-1}	0.1	Tenth		1795
centi	c	$1000^{-2/3}$	10^{-2}	0.01	Hundredth		1795
milli	m	1000^{-1}	10^{-3}	0.001	Thousandth		1795
micro	μ	1000^{-2}	10^{-6}	0.000 001	Millionth		1960
nano	n	1000^{-3}	10^{-9}	0.000 000 001	Billionth	Milliardth	1960

Unidades de Medida

Byte (armazenamento de dados)

- 1 Byte = 8 bits (dígitos `0` ou `1`)
- Imagem tons de cinza: 1 Byte por ponto (pixel)
- Caracteres de texto: 1 a 2 Bytes por caractere (depende da codificação)
- Números: inteiro (`int`: 4 Bytes, `long`: 8 Bytes), real (`float`: 4 Bytes, `double`: 8 Bytes)
- Disco rígido (HD) de 8TB, Memória RAM de 16 GB

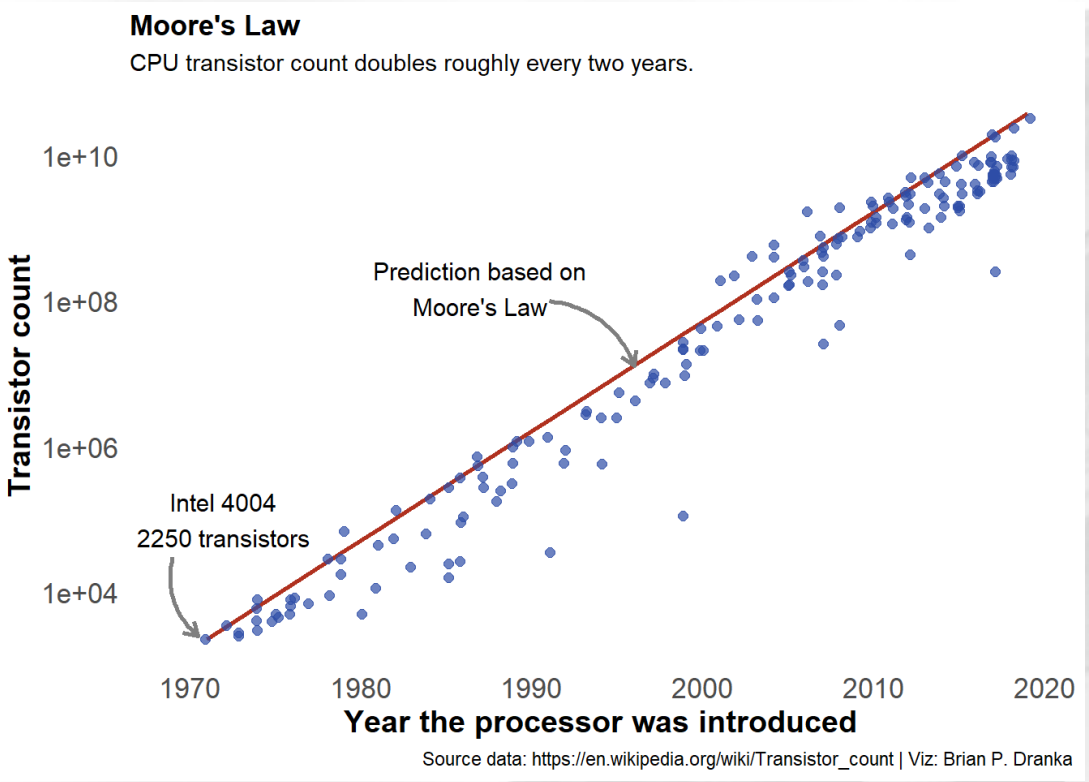
bps (bits por segundo)

- Velocidade de transmissão de dados (rede de 1 Mbps)

FLOP/s (Float Operations por segundo)

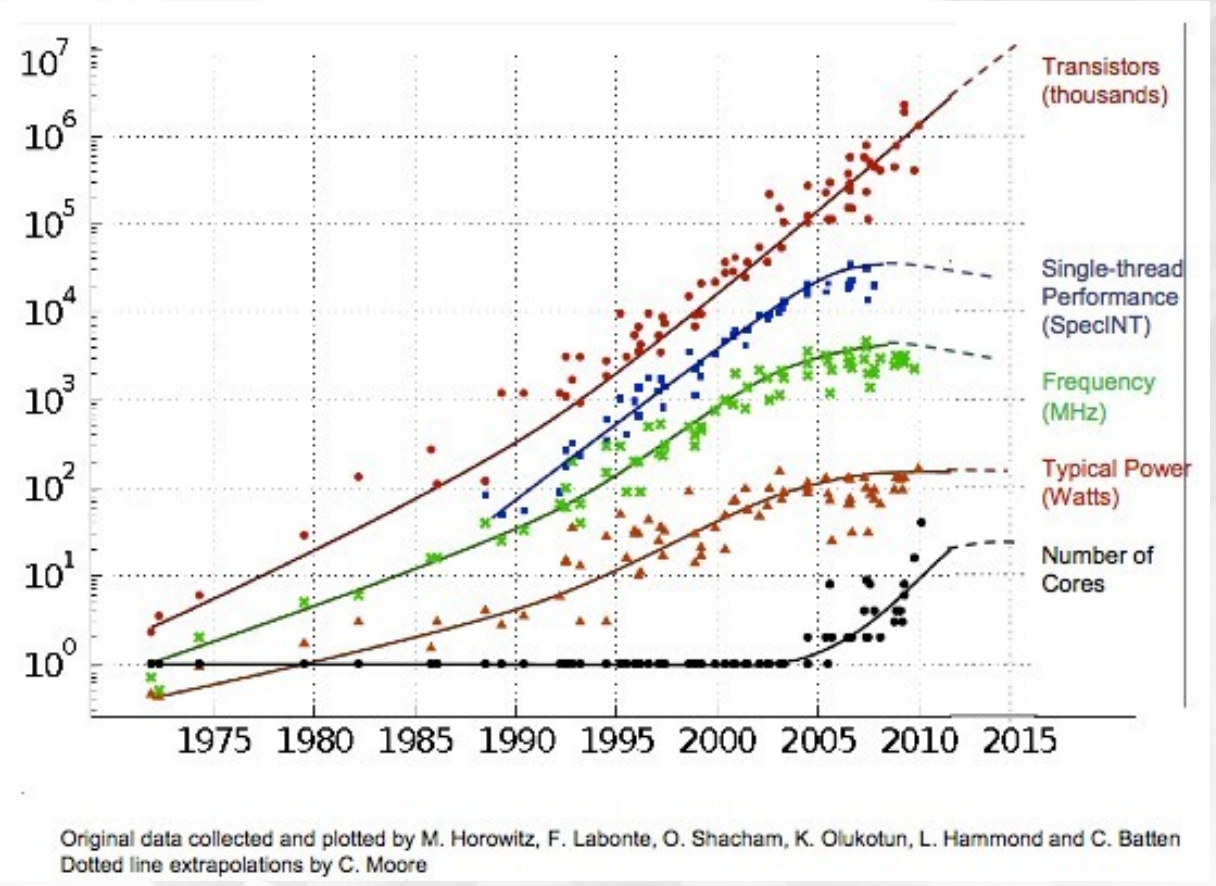
- Velocidade de operações aritméticas

Lei de Moore

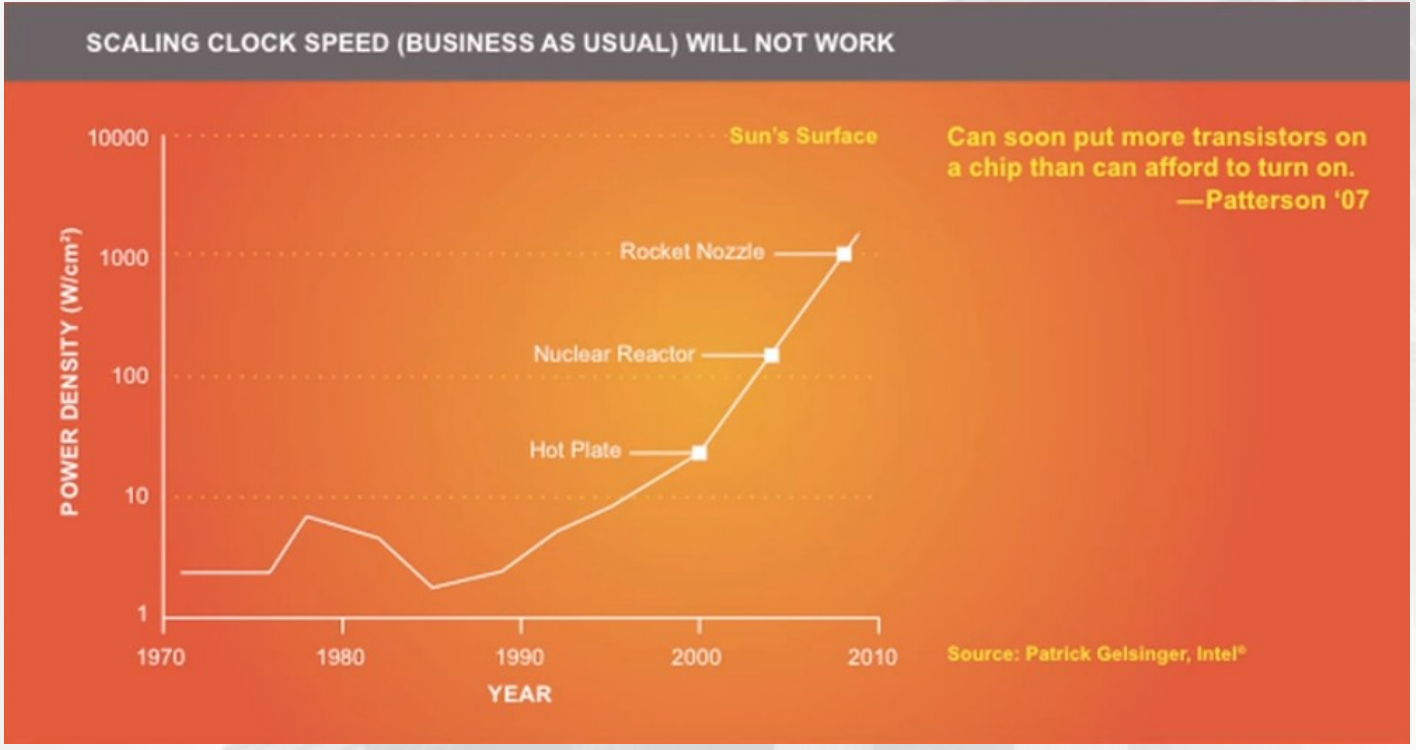


1965, G. Moore: número de transistores por chip duplica a cada 12-14 meses

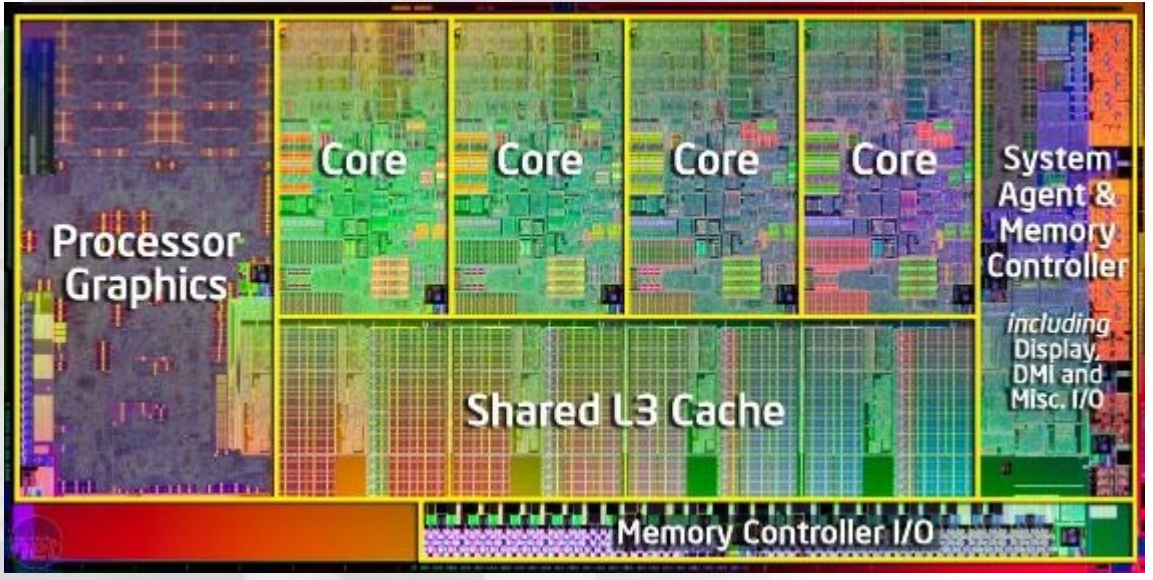
Lei de Moore


10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1975 1980 1985 1990 1995 2000 2005 2010 2015
Transistors (thousands)
Single-thread Performance (SpecINT)
Frequency (MHz)
Typical Power (Watts)
Number of Cores
Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

O problema do calor



Computadores para HPC



Computadores para HPC

Core

- Cada núcleo de uma pastilha. É a unidade básica de computação.
- Podem efetuar algumas operações aritméticas em paralelo

Nodo

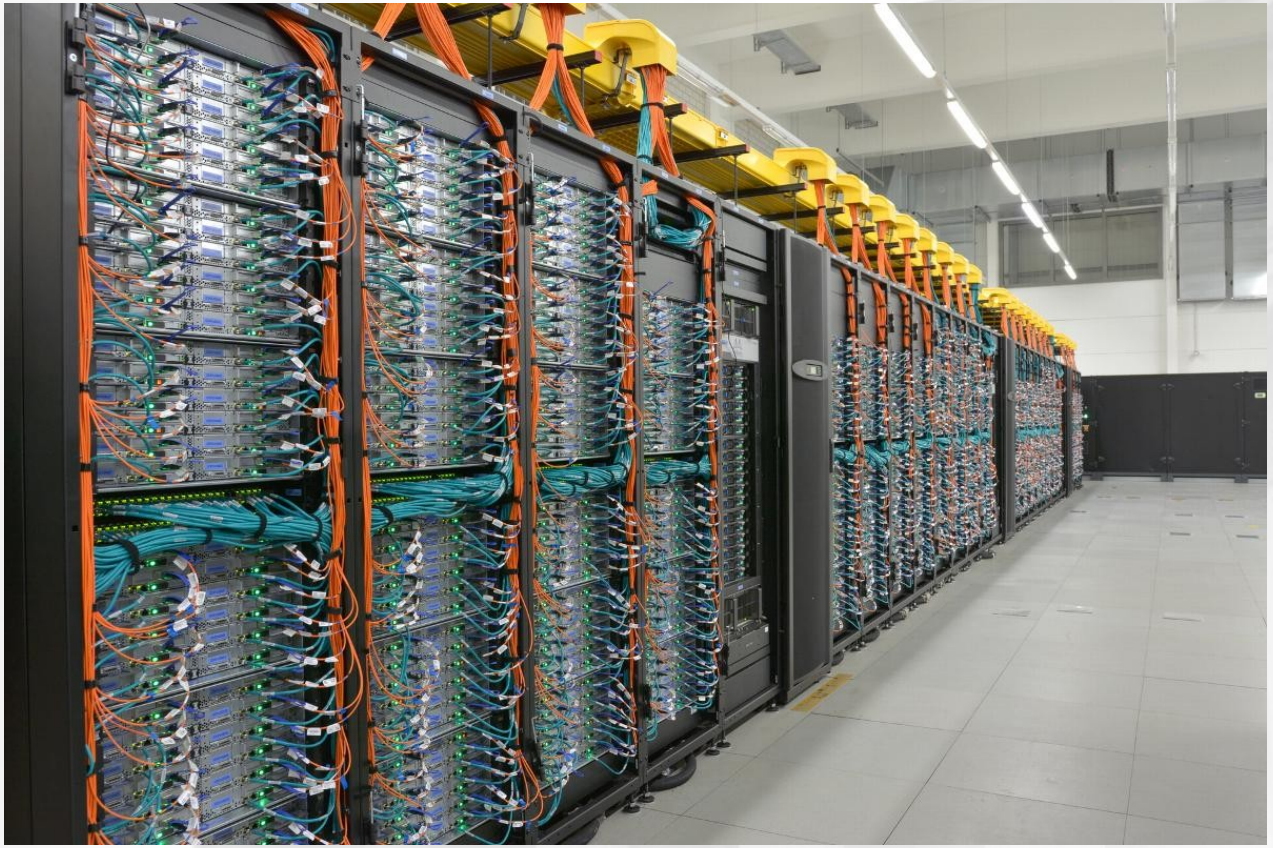
- Possui diversas pastilhas (CPU) combinadas em uma placa mãe
- Compartilham memória entre cores e entre pastilhas
- Pastilhas tem de 8 a 64 cores
- Aceleradores
- Troca de dados entre cores de uma mesma pastilha é rápida

Cluster

- Milhares de nodos conectados por uma rede de alta velocidade
- Comunicação entre nodos implica no envio de mensagens
- Alta latência, banda estreita

Computadores para HPC

Computadores para HPC



Sistema Operacional para HPC

GNU/Linux

SO para HPC

Nodo de acesso e nodos de processamento

Sistema de gerenciamento de trabalhos (*jobs*)

- Alocação de programas e usuários nos diversos nodos
- Controla tempo de execução e recursos/usuários
- Ex: slurm, pbs

Sistemas de arquivo paralelo

- Acesso simultâneo e paralelo
- Escalabilidade e redundância a falhas

Perguntas?

Armazenamento de Dados

Hard Disk Drive

Solid State Drive

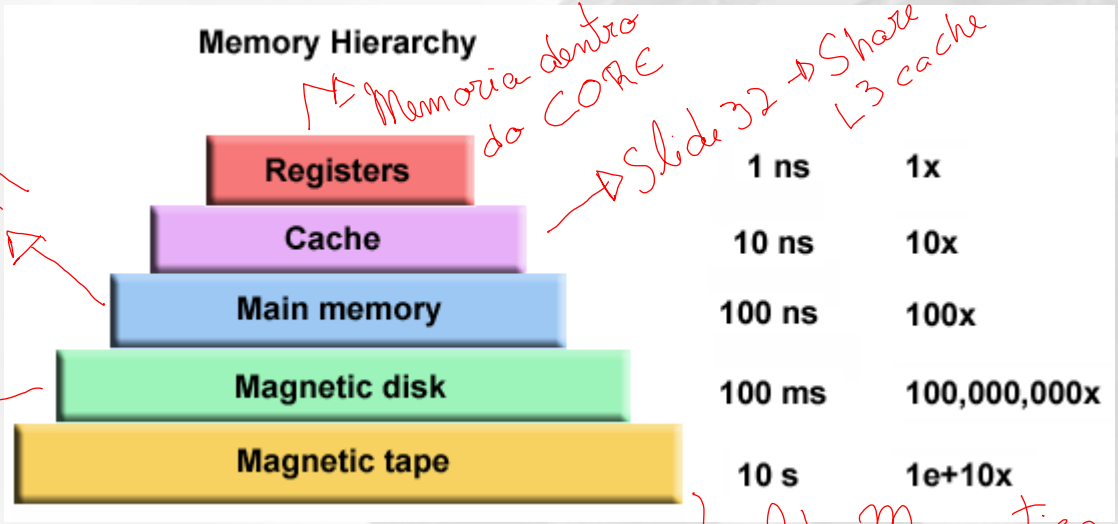
https://en.wikipedia.org/wiki/Solid-state_drive

Armazenamento de Dados

Volátil x não volátil

Acesso aleatório x sequencial

Se Reutilizar a informação
Salva em memoria SEMPRE
será MUITO mais RAPIDO



Redundant Array of Inexpensive Disks (RAID)

Combinação de dois ou mais discos

Padrões de organização, ou níveis:

- RAID 0: *stripping*
- RAID 1: espelhamento
- RAID 6: *stripping* de blocos com redundância de 2 discos
- RAID 1+0 ou 10: combinação

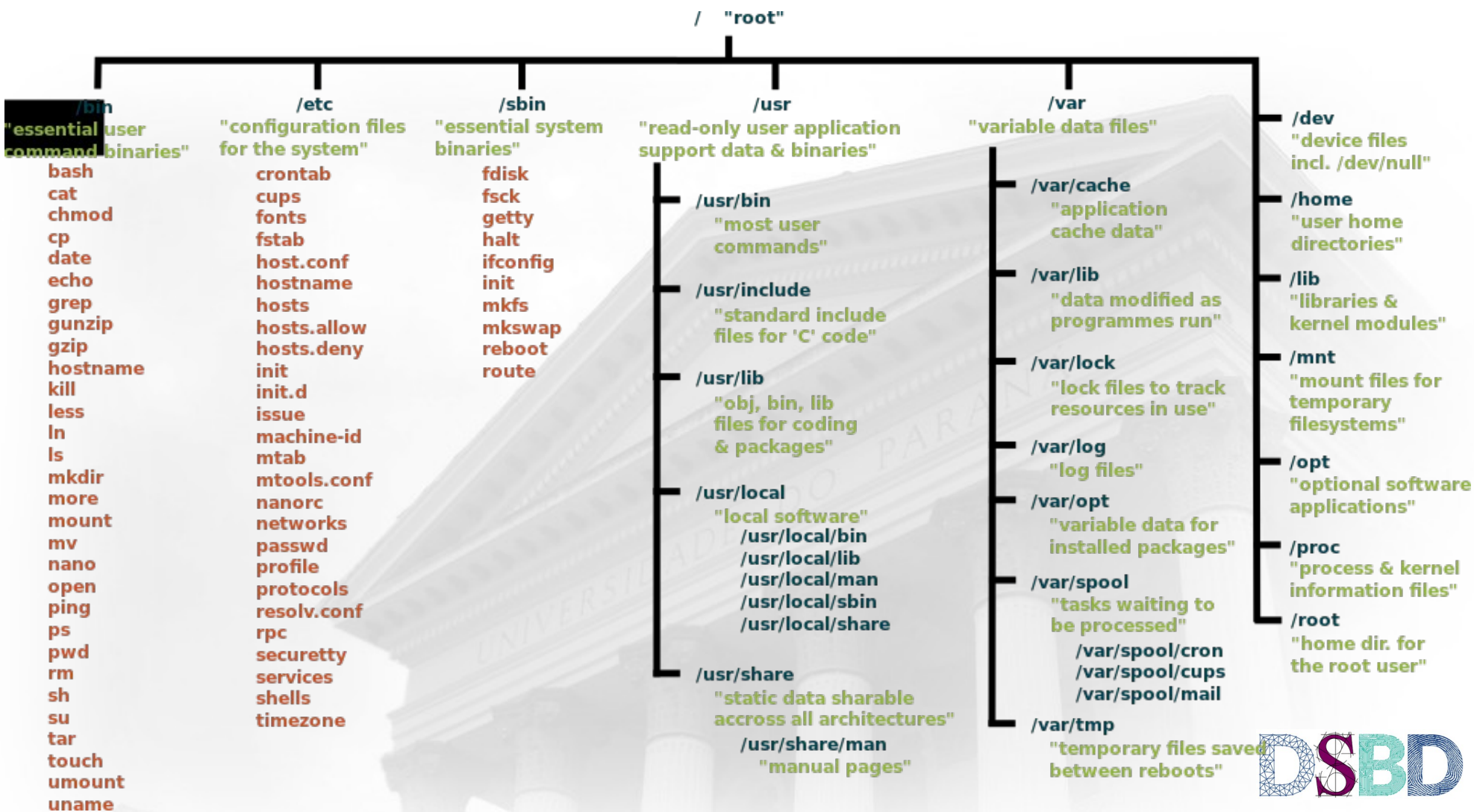
Implementação via software (md) ou hardware (controladora)

Sistemas de Arquivo

Sistemas de Arquivo

Virtual File System (VFS)

- Camada do Linux que permite acesso uniforme a diversos sistemas de arquivo
- Especifica uma interface (API) de acesso a arquivos. Padrão POSIX (open, close, read, write, seek, link, …)
- 2 conceitos fundamentais: arquivos e diretórios
- Estrutura de diretórios em árvore, com diretório raiz “/”
  - . : referência ao próprio diretório
  - .. : referência ao diretório pai


/ "root"
/bin
"essential user command binaries"
bash
cat
chmod
cp
date
echo
grep
gunzip
gzip
hostname
kill
less
ln
ls
mkdir
more
mount
mv
nano
open
ping
ps
pwd
rm
sh
su
tar
touch
umount
uname
/etc
"configuration files for the system"
crontab
cups
fonts
fstab
host.conf
hostname
hosts
hosts.allow
hosts.deny
init
init.d
issue
machine-id
mtab
mtools.conf
nanorc
networks
passwd
profile
protocols
resolv.conf
rpc
securetty
services
shells
timezone
/sbin
"essential system binaries"
fdisk
fsck
getty
halt
ifconfig
init
mkfs
mkswap
reboot
route
/usr
"read-only user application support data & binaries"
/usr/bin
"most user commands"
/usr/include
"standard include files for 'C' code"
/usr/lib
"obj, bin, lib files for coding & packages"
/usr/local
"local software"
/usr/local/bin
/usr/local/lib
/usr/local/man
/usr/local/sbin
/usr/local/share
/usr/share
"static data sharable accross all architectures"
/usr/share/man
"manual pages"
/var
"variable data files"
/var/cache
"application cache data"
/var/lib
"data modified as programmes run"
/var/lock
"lock files to track resources in use"
/var/log
"log files"
/var/opt
"variable data for installed packages"
/var/spool
"tasks waiting to be processed"
/var/spool/cron
/var/spool/cups
/var/spool/mail
/var/tmp
"temporary files saved between reboots"
/dev
"device files incl. /dev/null"
/home
"user home directories"
/lib
"libraries & kernel modules"
/mnt
"mount files for temporary filesystems"
/opt
"optional software applications"
/proc
"process & kernel information files"
/root
"home dir. for the root user"

Sistemas de aquivo de disco

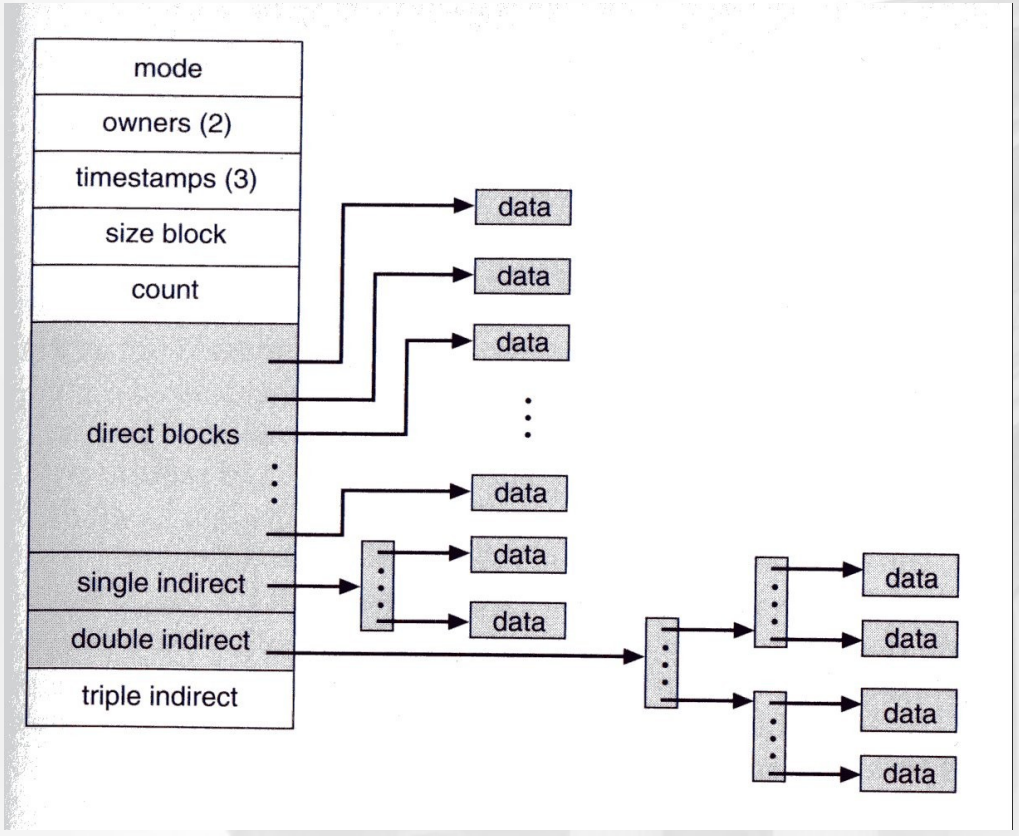
Gerencia os blocos de dados (setores) do disco

- Associa nomes de arquivos a blocos
- Mantém metadados
- Controla espaço livre (fragmentação), quota, permissões

Diversas implementações

- ext4, zfs, ntfs, fat
- Algumas implementações fazem versionamento ou *journaling*

Sistemas de Arquivo em disco



Partições

Partições proveem uma melhor separação dos dados em disco

- Cada partição tem seu próprio sistema de arquivos
- Comando **`fdisk, parted`**
- Partição de dados x partição de *swap*
- Segurança para falha no sistema de arquivos

Sistemas de Arquivo

Um SA precisa ser montado antes de ser acessado

- Montar significa indicar o diretório a partir do qual o SA será acessado neste computador
- A montagem sobrepõe qualquer dado existente (fica inacessível)
- **`df -h`**
- **`mount`**

Sistemas de Arquivo em Rede

Um SA pode ser exportado pela rede e montado por diversos clientes

- Arquitetura cliente-servidor
- Compartilhamento se dá através da montagem via protocolos específicos
- NFS, SAMBA, DNDB3

Clustered File System é um SA que distribui os dados em diversos servidores

- Redundância a falhas, acesso paralelo, escalabilidade
- Geralmente baseados em objetos (*object file system*):
  - Separação de Metadados e Dados
- GFS, Lustre, Hadoop, Gluster

Sistemas de Arquivo especiais

`/dev`

- Acesso direto aos dispositivos (*devices*) da máquina.
- **`ls -l /dev/sd*`**

`/proc`

- Acesso aos processos
- **`cat proc/cpuinfo`**

`/sys`

- Acesso aos dispositivos através do kernel. Muito utilizado para configurações
- **`cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`**

Arquivos

“Em um sistema UNIX, tudo é um arquivo; se algo não é um arquivo, é um processo”

Tipos de arquivo (opção `-l` do comando `ls`)

- **Arquivos regulares (`-`):** armazenam dados. Não há divisão em nome.extensão
- **Diretórios (`d`):** são arquivos que contém outros arquivos
- **Dispositivos (`b`, `c`):** acessam dispositivos de hardware. Podem ser arquivos de bloco ou caracteres
- **Links (`l`):** ponteiros para outros arquivos. Podem ser *soft* ou *hard links*.

Sistema de Arquivos Linux

Caminhos

- **`$PATH`**
- caminhos relativos (`.` e `..`) e absolutos (`/`)

Diretório HOME

- **`quota -vs`**
- **`~`**

Segurança de arquivos

GNU/Linux tem um sistema bastante rígido de permissões para arquivos

- Todo arquivo pertence a um usuário e um grupo
- As permissões de leitura, escrita, e execução devem ser definidas para o usuário, grupo e outros
- Comando **`id`** mostra usuário e grupos aos quais pertence
- Comando **`ls -l`** mostra permissões de forma posicional
  - usuário, grupo, outros

Modos de acesso para arquivos

Código	Significado
0 ou **-**	Acesso desta posição não concedido
1 ou **x**	Permissão de execução nesta posição
2 ou **w**	Permissão de escrita nesta posição
4 ou **r**	Permissão de leitura nesta posição
u	Permissão do usuário
g	Permissão do grupo
o	Permissão para outros
a	Permissão para todos

Cuidados no uso de arquivos

Acesso a arquivos é várias ordens de grandeza mais lento que o processador

Performance é dominada pelo número de acessos a disco

- ~10 ms por acesso

Custo do acesso é dominado pela latência

- *tempo de busca + latência de rotação + bytes / bandaDisco*
  - *1 setor: 5ms + 4ms + 2,5µs (≈ 512 B/200 MB/s) ≈ 9ms*
  - *100 setores: 5ms + 4ms + 0,25ms ≈ 9,25ms*
  - *100 vezes mais dados com 3% de aumento no tempo*

Perguntas?

Acesso local ao Laboratório

Comandos Básicos

Bash

Bash é um interpretador de comandos. É um programa usado para iniciar e controlar a execução de outros programas.

- Possui uma sintaxe própria para programação
- Define alguns comandos internos (cd, exit, logout, pwd)
- Define algumas variáveis de ambiente (HOME, PATH, PS1)

Obtendo ajuda

GNU/Linux tem a filosofia de tornar seu usuário mais independente.

- Diversos fóruns ajudam com perguntas
- Em geral, assume-se que o usuário leu o manual antes
  - Comandos: **`man, info, whatis, apropos`**
    - Teclas de navegação: **`/string`** (busca), **`q`** (para encerrar)
  - Opção **`--help`**
  - **`RTFM!`** é uma resposta comum a perguntas cuja resposta está no manual

Comandos iniciais

Comando	Significado
`ls`	mostra a lista de arquivos de um diretório
`cd <diret>`	muda de diretório corrente
`less <arq>`	mostra o conteúdo de um arquivo
`cat <arqtxt>`	mostra o conteúdo do arquivo `<arqtxt>`
`pwd`	mostra o diretório corrente
`exit` ou **`logout`**	sai da seção atual
`man` ***`comando`***	ler páginas de manual sobre **comando**
`apropos` ***`string`***	procura pela **string** na base do *whatis*

Combinações de tecla em Bash

Tecla(s)	Função
`Ctrl+c`	encerra a execução de um programa
`Ctrl+d`	encerra a seção atual do shell
`Ctrl+l`	limpa a tela
`Ctrl+r`	procura no histórico de comandos
`Ctrl+z`	suspende um programa
`SetaCima/Baixo`	navega no histórico de comandos
`Shift+PageUp/ Shift+PageDown`	navega no *buffer* do terminal (para ver texto que passou)
`Tab`	completa comando ou nome de arquivo
`Tab Tab`	mostra opções de comandos ou arquivos

Exercícios

Digite os comandos a seguir e tente interpretar o que acontece. Pergunte!

`echo hello world`	`whoami`	`echo $SHELL`
`date`	`who`	`echo {con,pre}{sent,fer}{s,ed}`
`hostname`	`id`	`man ls (q)`
`arch`	`last`	`cal 2018`
`uname -a`	`finger`	`echo 3*5 | bc -l`
`dmesg | less`	`w`	`yes please (Ctrl+c)`
`uptime`	`file .`	`time sleep 5`
`echo $HOME`	`top (q)`	`history`

Manipulando Arquivos

Manipulação de permissões

Comando	Significado
`chmod`	modifica as permissões de um arquivo
`chown`	modifica usuário ou grupo de um arquivo
`mkdir`	cria um diretório
`cp -R -avu`	copia um arquivo
`mv`	move um arquivo
`rm -f -r <dir>`	apaga o diretório <dir>
`head` ou **`tail`**	mostra linhas iniciais ou finais do arquivo
`ln -s`	faz um link entre arquivos

Arquivos, usuários e permissões

Comando	Significado
`finger`	mostra informações sobre usuário
`who`	mostra usuários logados no sistema
`quota`	mostra a quota do usuário
`find`	procura arquivos

Manipulação de arquivos

Comando	Significado
`ls -a -l -R -F -t`	mostra a lista de arquivos de um diretório
`file <arq>`	mostra o tipo do arquivo
`mkdir`	cria um diretório
`cp -R -avu`	copia um arquivo
`mv`	move um arquivo
`rm -f -r`	remove um arquivo
`head` ou `tail`	mostra linhas iniciais ou finais do arquivo
`ln -s`	faz um link entre arquivos
`touch`	muda a data de um arquivo

Wildcards

São caracteres especiais usados para criar padrões definindo um conjunto de arquivos ou diretórios

- * - representa zero ou mais caracteres
- ? - representa apenas um caractere
- [ ] - representa um intervalo de caracteres
- [^] - representa a negação de um intervalo de caracteres

Wildcards

Comando	Significado
`ls b*`	Arquivos iniciando com **b**
`ls -ld .g*`	Arquivos iniciando com **.g**
`ls ?i*`	Arquivos com um caractere seguido de **i**
`ls [sv]*`	Arquivos que iniciem com **s** ou **v**
`ls *[0-9]*`	Arquivos com um dígito de **0-9**
`mv *.??g fotos/`	Move arquivos .png e .jpg para dir. fotos
`ls *[[:upper:]]*`	Arquivos com uma letra maiúscula

Referências

- Anatomy of the Linux kernel
- Linux OS Tutorial
- Introduction to UNIX
- Introduction to Linux