

Big Data & Data Science

Infraestrutura Computacional **Parte 1: Linux e Shell**



Introdução

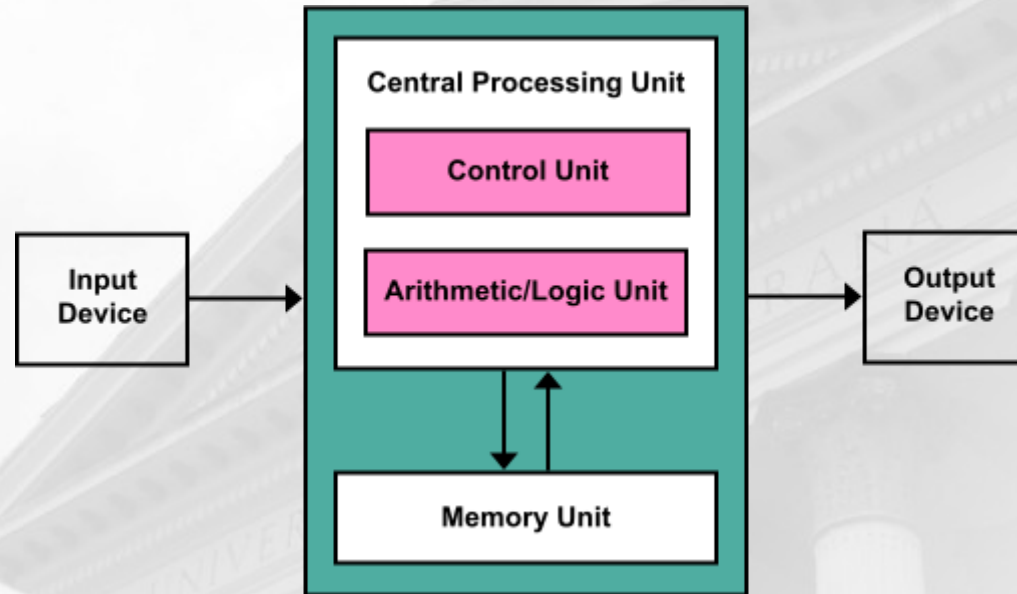


Apresentação

Prof. Daniel Weingaertner

- ▶ danielw@inf.ufpr.br
- ▶ web.inf.ufpr.br/danielw
- ▶ Áreas de Pesquisa
 - ▶ Computação Científica
 - ▶ Processamento de Imagens
 - ▶ Humanidades Digitais

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

- ▶ **GNU's Not Unix**: 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

Licença 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

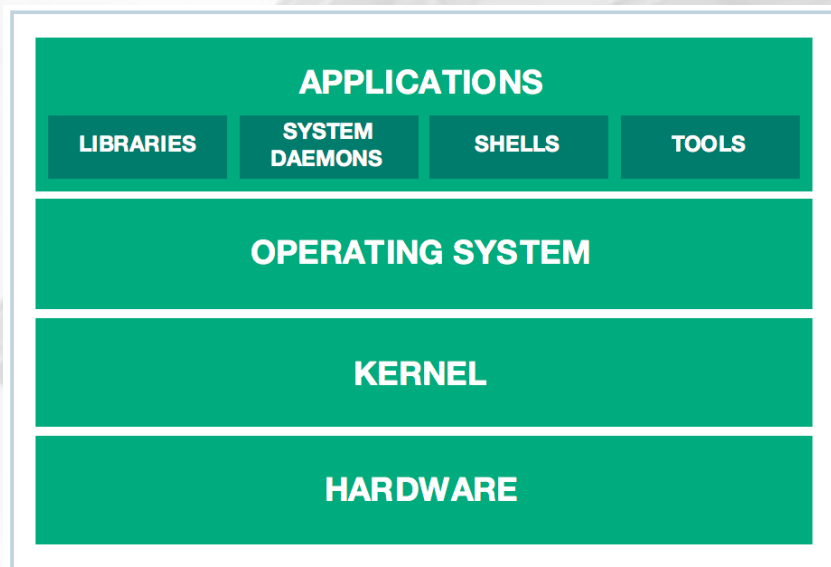
Características do GNU/Linux

- ▶ **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

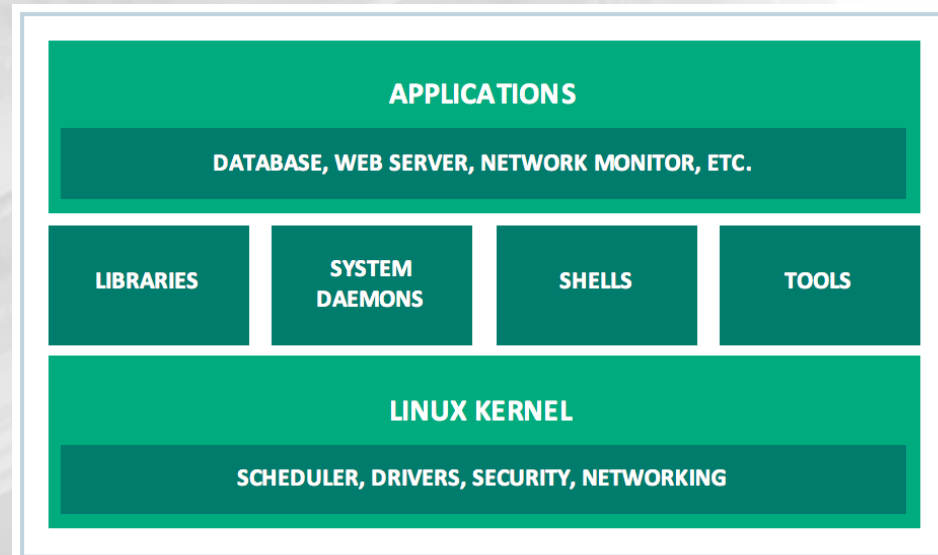
O Sistema Operacional é um software que controla o hardware e faz a interface deste com as aplicações



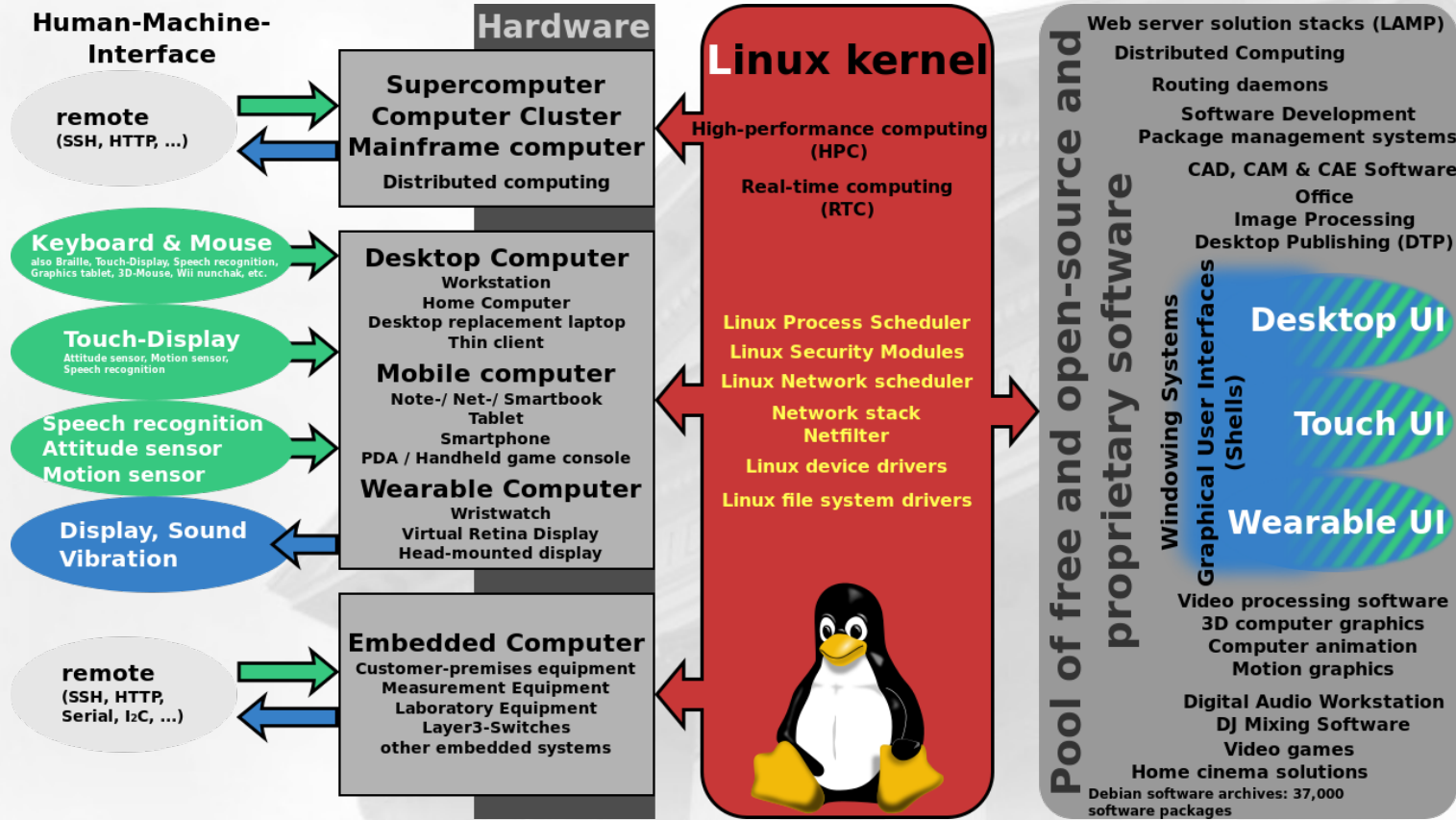
Sistema Operacional GNU/Linux

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

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



Sistema Operacional GNU/Linux



Kernel Linux

O kernel é uma parte do SO:

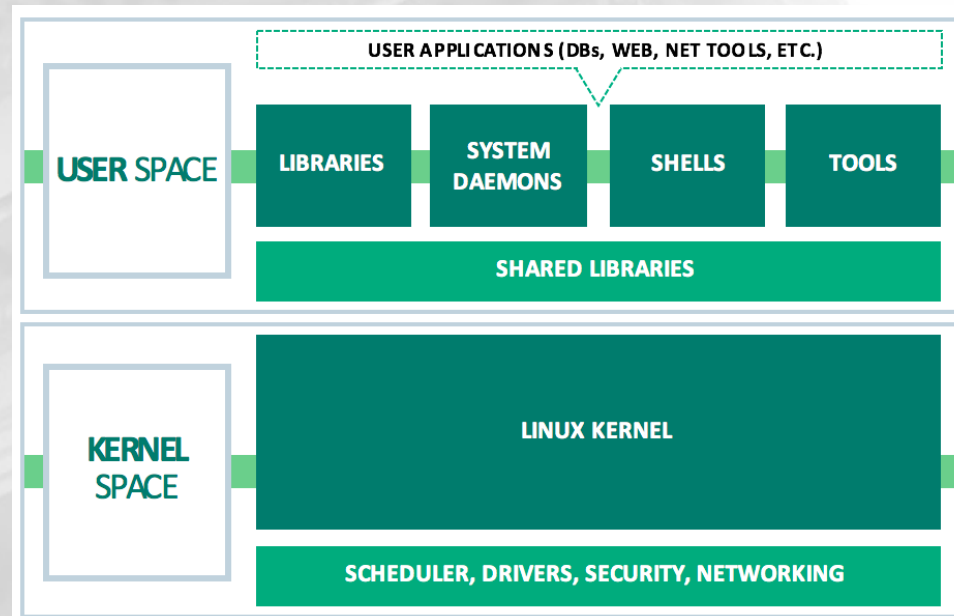
- ▶ Controla a CPU, memória e outros dispositivos
- ▶ Acessa dados em dispositivos de armazenamento
- ▶ Escalona processos
- ▶ Roda aplicações, isolando-as umas das outras
- ▶ Disponibiliza uma API (*system calls*) para atividades restritas

User × Kernel space

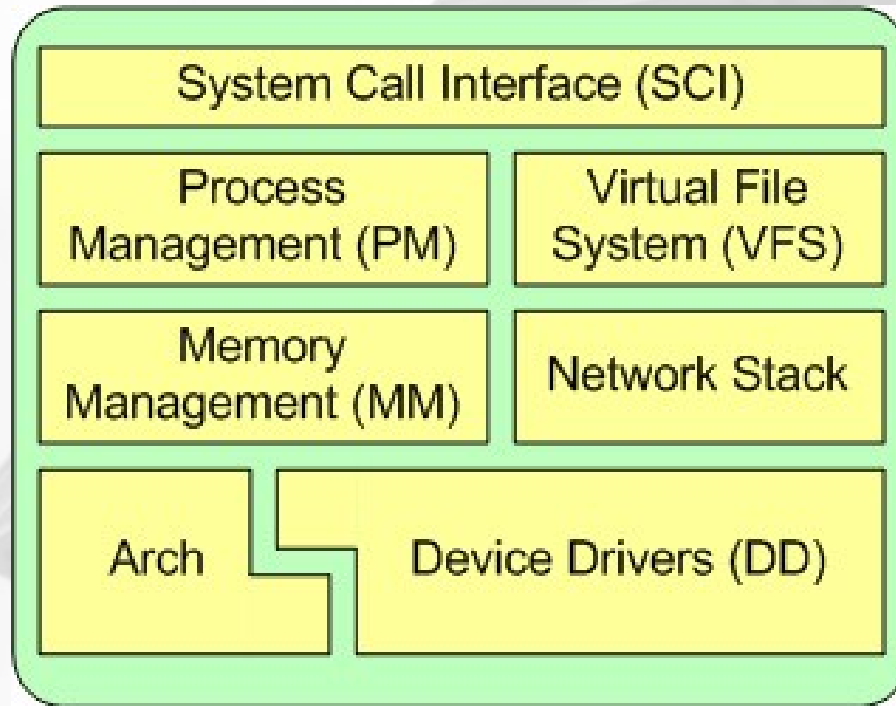
O GNU/Linux executa seu kernel (Linux) em uma região de memória restrita e protegida (*kernel space*)

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

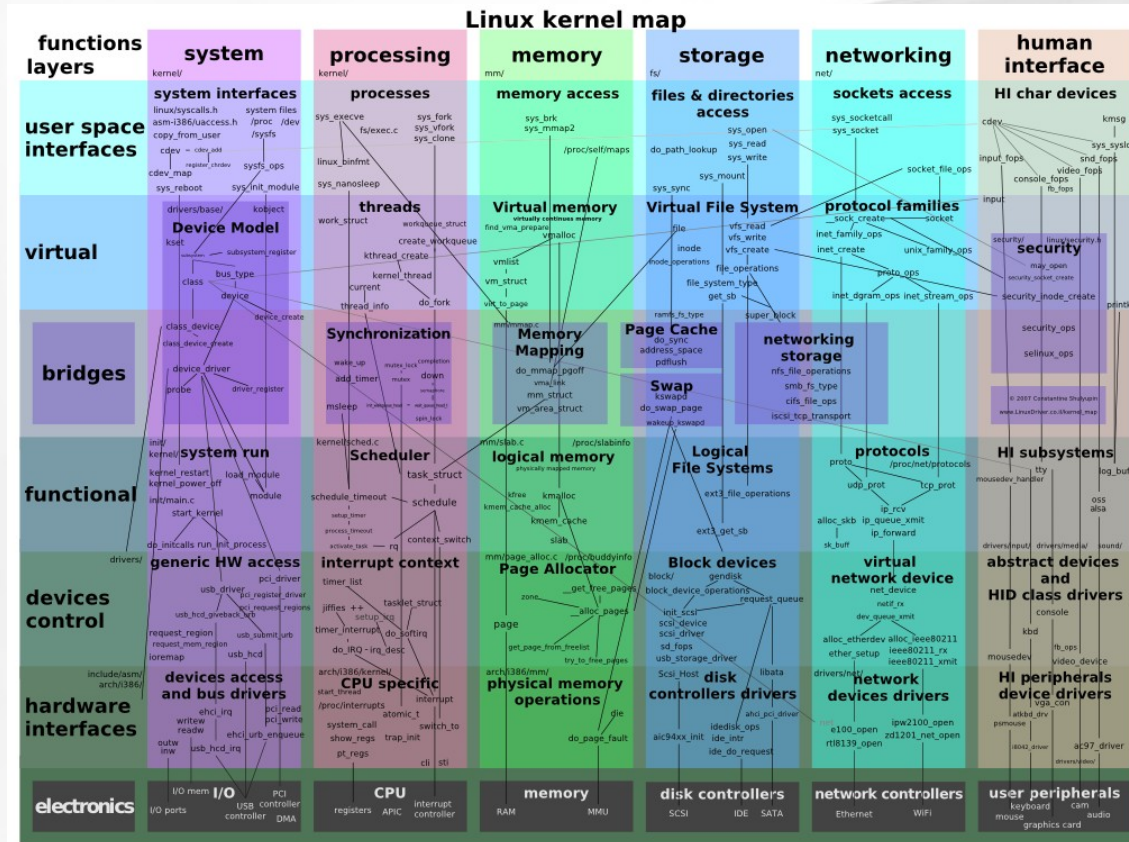
► Spectre & Meltdown



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

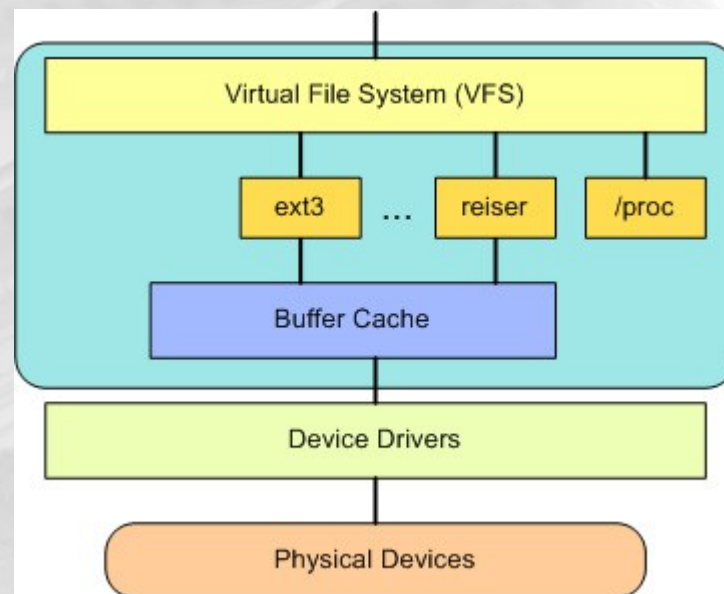
Gerenciamento de Memória (MM)

- ▶ 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



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)

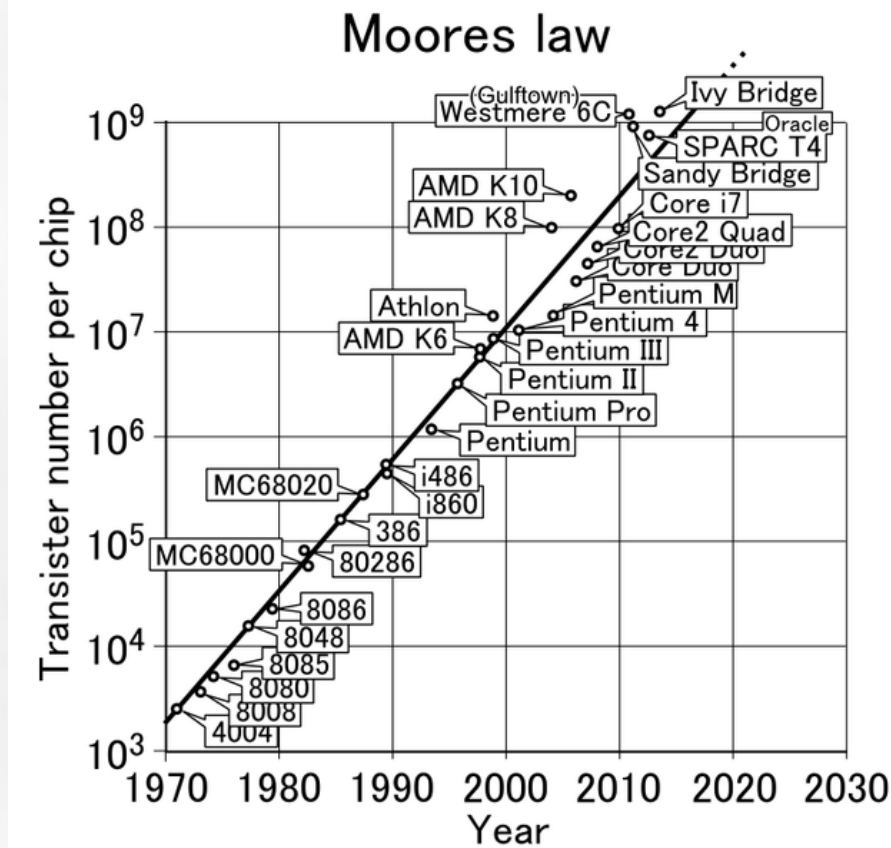
- ▶ Velocidade de transmissão de dados (rede de 1 Mbps)

FLOP/s (Float Operations por segundo)

- ▶ Velocidade de operações aritméticas

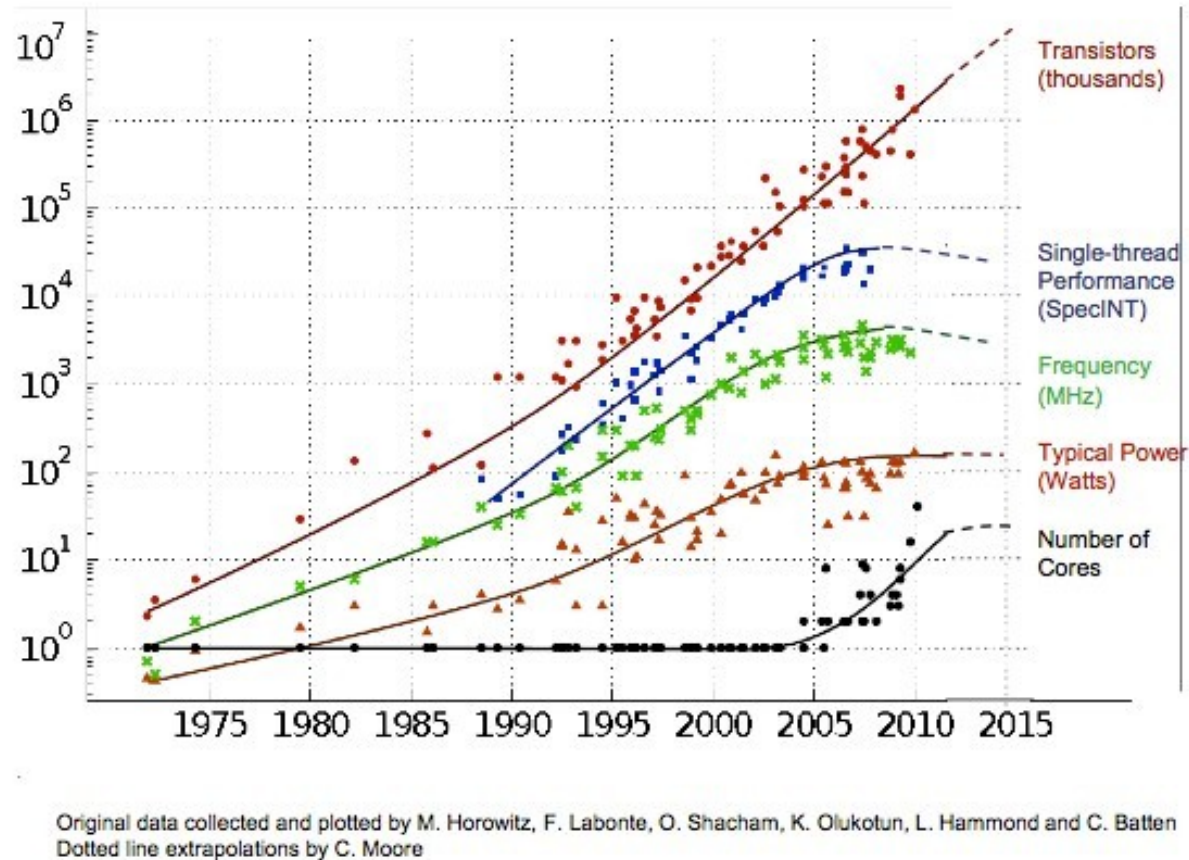


Lei de Moore

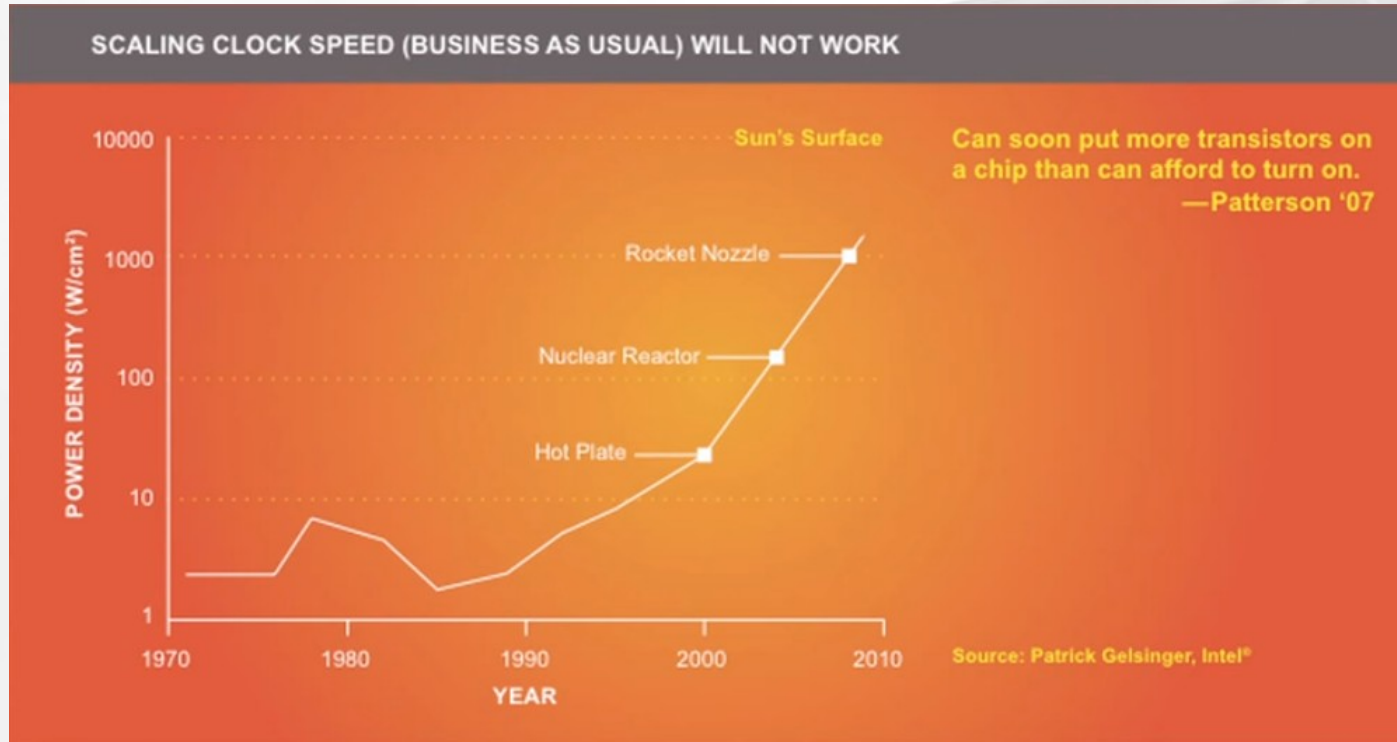


1965, G. Moore: número de transistores por chip duplica a cada 12-14 meses

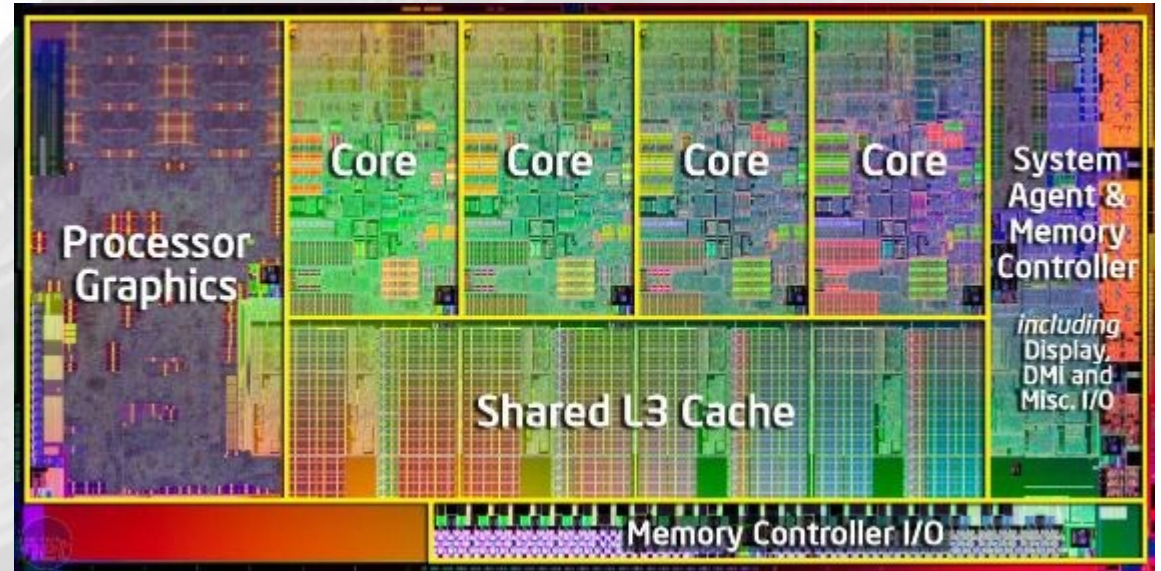
Lei de Moore



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?

Acesso local ao Laboratório

Login no sistema do DInf

Interface GNOME

- ▶ Login local nos terminais

Troca de senha

- ▶ O que é uma boa senha?
- ▶ Abrir um Terminal de Comando
 - ▶ `passwd`

Acesso Remoto através de um terminal

- ▶ `ssh ssh@inf.ufpr.br`

Acesso ao Material do Curso

Sistema Moodle

- ▶ moodle.c3sl.ufpr.br
- ▶ Curso: Infraestrutura Computacional – Parte 1
 - ▶ senha: dsdb18

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 <i>comando</i>	ler páginas de manual sobre comando
apropos <i>string</i>	procura pela string na base do <i>whatis</i>

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 <i>buffer</i> 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!

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



Referências

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

▪