

IOF0265 - Técnicas de Visualização e Distribuição de Dados Oceanográficos

Aula 1 - Linux

Paulo S. Polito, Ph.D. e Olga T. Sato, Ph.D.
`olga.sato@usp.br` & `polito@usp.br`

Instituto Oceanográfico da Universidade de São Paulo

1

Noções Básicas

- Partes do Computador
- Software Básico
- Interagir com o Computador
- Binário, Decimal, Hexadecimal, ASCII

2

Exemplo de formato de dados

- NETCDF

Roteiro

1

Noções Básicas

- Partes do Computador
- Software Básico
- Interagir com o Computador
- Binário, Decimal, Hexadecimal, ASCII

2

Exemplo de formato de dados

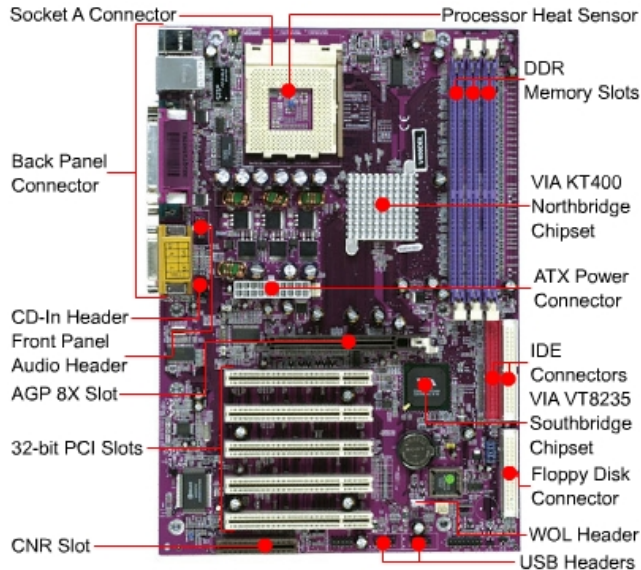
- NETCDF

O Modelo de von Neumann (1940)



- **CPU**: Unidade Central de Processamento
 - Unidade de Controle,
 - Unidade de Aritmética e Lógica (ALU),
 - Registradores,
 - Input/Output básico.
- **Input**: rede, teclado, leitor de DVD, mouse, scanner.
- **Output**: rede, impressora, monitor, gravador de CD, DVD.
- **Memória volátil**: RAM, cache.
- **Memória permanente**: HD, USB-drive, mídia de CD/DVD.

Na Prática...



Além da placa mãe, no gabinete vão:

- HD, SSD, DVD.
- Coolers.
- Fonte 110V ~ 3.5, 5, 12V.
- Cabos, conectores e jumpers.
- Placas de vídeo, audio, rede etc.

Três partes da CPU

UC: Unidade de controle: organiza e processamento, **como um maestro.**

ALU: Unidade de aritmética e lógica: faz as contas, **como os músicos.** Que contas são essas?

$+$	$-$	\times	\div	$=$	$>$	$<$	\neq	$\&$	$ $
-----	-----	----------	--------	-----	-----	-----	--------	------	-----

 Só isso.

Memórias: Também há três tipos:

- Registradores: guardam os números que serão usados nas contas e endereços de memória;
- Acumuladores: guardam o resultado das contas;
- Cache: intermedia dados entre a CPU e a memória RAM.

Há memórias e memórias

Os dados ficam na CPU apenas durante as contas. Usamos memórias que ficam fora da CPU:

- **Volátil** (i.e. perde se desligar): RAM.
- **Permanente** (i.e. não perde): HD, DVD, SSD, pen-drive.

Em termos relativos, as memórias

- RAM são pequenas, caras e rápidas;
- SSDs são médios, razoáveis e rápidos;
- HDs são grandes, baratos e lentos.

O Sistema Operacional



- O sistema operacional controla:
 - Execução de tarefas automáticas (e.g.: conexão wifi)
 - Execução de programas em geral (e.g.: browser, jogos)
 - Controle de periféricos (e.g.: acesso a DVD, impressora)
- É composto de diversos programas:
 - **Kernel** é o principal.
 - Redes (FTP, SSH, daemons)
 - Interface gráfica (janelas, docas, ícones)
 - Segurança (firewall), etc.

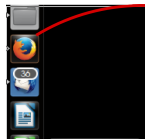
POSIX: Portable Operating System Interface

- O padrão POSIX garante que um grande conjunto de **comandos** funcione em vários sistemas operacionais.



- OS X (Mac)
- Linux
- FreeBSD
- Android
- IRIX (SGi)
- HP-UX (HP)
- Solaris (Sun)

Comandos × “point & click”



Interface Gráfica (Unity,Gnome,KDE)

Arquivo de configuração

```
...  
Exec=/usr/lib/firefox/firefox %u  
Icon=Firefox  
...
```

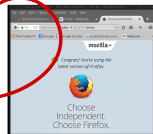
Sistema Operacional

Lê arquivo de configuração

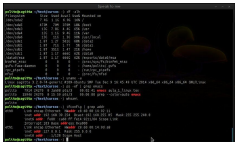
Executa o que estiver escrito depois de "Exec="

Interface Gráfica

Coloca moldura em volta da janela
Coloca ícones no canto da moldura



Comandos × “point & click” ... continuando



Sistema Operacional

Executa comando "firefox" do terminal

Interface Gráfica

Coloca moldura em volta da janela

Coloca ícones no canto da moldura



O Terminal é teu amigo

```
polito@sagitta:~/text/cursos:~) df -alh
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        7.4G  1.1G  6.0G  16% /
/dev/sda5        473M   79M  370M  18% /boot
/dev/sda3        13G   7.8G  4.4G  65% /usr
/dev/sda4        12G   1.1G  9.4G  11% /var
/dev/sda6        13G   11G   1.3G  90% /usr/local
/dev/sde1        1.8T  1.2T  561G  68% /data2
/dev/sdd1        1.8T   71G  1.7T   5% /data1
/dev/sdb1        1.8T  351G  1.4T  21% /home
/dev/sdc1        1.8T  1.1T  666G  62% /data0
/data0/esa       1.8T  1.1T  666G  62% /exports/data0/esa
binfmt_misc      0      0      0    - /proc/sys/fs/binfmt_misc
gvfs-fuse-daemon  0      0      0    - /home/polito/.gvfs
rpc_pipefs       0      0      0    - /run/rpc_pipefs
nfsd              0      0      0    - /proc/fs/nfsd

polito@sagitta:~/text/cursos:~) uname -a
Linux sagitta 3.2.0-74-generic #109-Ubuntu SMP Tue Dec 9 16:45:49 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux

polito@sagitta:~/text/cursos:~) ps -ef | grep enacs
polito    7914 24270  0 Jan08 pts/3    00:02:41 enacs aula_1_linux.tex
polito    15946 24270  0 15:59 pts/3    00:00:00 grep --color=auto enacs

polito@sagitta:~/text/cursos:~) whoami
polito

polito@sagitta:~/text/cursos:~) ifconfig | grep addr
eth0      Link encap:Ethernet HWaddr c8:60:00:14:92:11
          inet addr:192.168.39.254 Bcast:192.168.255.05 Mask:255.255.240.0
          inet6 addr: fe80::ca60:ff:fe14:9211/64 Scope:Link
          Interrupt:103 Base address: 0xa000
eth1      Link encap:Ethernet HWaddr c8:60:00:14:9d:a8
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host

polito@sagitta:~/text/cursos:~) █
```

O Terminal é teu amigo ... continuando

- Rápido.
- Informativo (date, df, pwd, ls, ps -ef).
- Interativo (detecta erros, manda mensagens).
- Não esconde o que está acontecendo.
- Facilita testes, aprendizado (man).
- Induz a automação de processos.

O Terminal é teu amigo e fala BASH.

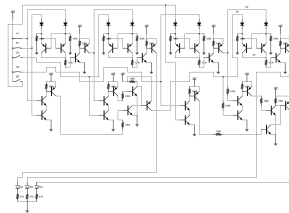
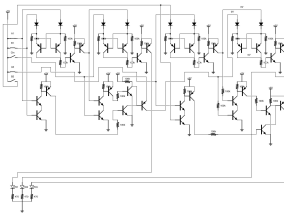
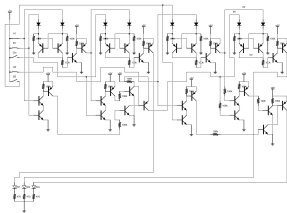
- Chamamos de Shell as linguagens interpretadas para interagirmos com o sistema operacional.
- Serve para tarefas simples, e.g.:
 - Executar o mesmo programa em vários arquivos.
 - Automatizar processos repetitivos.
 - Executar sequências de programas.
- Há várias: SH, CSH, TCSH, BASH etc. A maioria das distribuições Linux usa BASH.
- O terminal - que é teu amigo - aceita comandos do sistema operacional, dos shells e outros.
- Tutorial: [http:// www.tldp.org/LDP/abs/html/](http://www.tldp.org/LDP/abs/html/)

O que fazer com BASH?

- Vamos estudar o `exe_pra_bash.bash`.
- Nele vamos manipular vários arquivos sem usar o mouse.
- Vamos renomear, editar e reformatar 10 arquivos.
- Vamos fazer contas simples (BASH não é bom nisso).
- Vamos executar esse programa e verificar se funcionou.
- **Pense nas possibilidades.**

Números e Símbolos

- No exemplo anterior processamos um arquivo de texto.
- Arquivos de dados oceanográficos não são necessariamente assim.
- **Porque?**



Como a máquina realmente opera?

- Exemplo: queremos somar dois decimais: $97 + 3$.
- Computadores fazem contas com circuitos feitos de transístores que tem duas posições: aberto (0) e fechado (1).
- Para fazer a soma precisamos colocar os números na memória que a CPU acessa.
- Para guardar o número 97 numa memória dentro da CPU fazemos: **10110000 01100001** em binário.

Como a máquina realmente opera? ... continuando

- O circuito da CPU é tal que quando recebe os bits 10110000 ele conecta 01100001 com uma memória chamada “registrador AL”.
- 10110000 01100001 é o mesmo que **B0 61** em hexadecimal.

bin.	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1
	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
	8+	0+	2+	1+	0+	0+	0+	0	0+	4+	2+	0	0+	0+	0+	1
	11				0				6				1			
hex.	B				0				6				1			

Como a máquina realmente opera? ... continuando

- Cada família de CPUs tem seu conjunto de códigos que dependem do hardware, essa é a *linguagem assembly*.
- Para o PC8086 **B0** é interpretado como **MOV AL**, i.e. MOVa o número **61** para o registrador chamado **AL**.
- 01100001 em binário é o mesmo que 61 em hexadecimal que é o mesmo que $(6 \times 16^1 + 1 \times 16^0) = (6 \times 16 + 1 \times 1) = 97$ em decimal.
- Note que até agora apenas colocamos 97 num registrador. A seguir precisaríamos de mais duas instruções:
 - somar 3,
 - ler o resultado de um acumulador.

Como a máquina realmente opera? ... continuando

- Muita complicação para uma operação simples? **Todos os computadores funcionam dessa forma.**
- Como seria muito complicado programar assim foram construídas linguagens de programação mais simples (e.g.: FORTRAN, C++, Java).
- A partir dessas linguagens compiladas desenvolveram-se linguagens interpretadas (e.g.: Matlab, Python, Bash), ainda mais fáceis.
- A ideia é aproximar a linguagem dos computadores da dos humanos.

Para guardar a informação

- Recapitulando: Todo dado é guardado na forma binária.
- Lidar diretamente com números binários é complicado.
- A solução se dá em dois passos:
 - o uso de hexadecimais para compactar a notação;
 - a conversão de códigos hexadecimais para letras, números e símbolos.
- Há vários padrões de conversão, e.g.:
 - ISO 8859-1 e Windows-1252: Caindo em desuso.
 - **ASCII**: Caindo em desuso, 1 byte (usa só 7 bits).
 - **UTF-8**: Mais popular, 1 a 4 bytes, contém ASCII.
- Por exemplo, em UTF-8 de 1 byte (i.e. ASCII):

Para guardar a informação ... continuando

Binário	Hexadecimal	Caracter
0011 0001	3 1	1
0101 0000	5 0	P
0101 0001	5 1	Q
0111 0001	7 1	q
0000 0111	0 7	BEL

- Portanto, quando um você salva um **texto** que contém o **caracter** “1” o computador escreve no disco “0011 0001”: 1 byte, 8 bits
- A informação relevante poderia ser salva em 1 bit.
- Salvar texto ocupa mais espaço do que salvar binário.**

Roteiro

1

Noções Básicas

- Partes do Computador
- Software Básico
- Interagir com o Computador
- Binário, Decimal, Hexadecimal, ASCII

2

Exemplo de formato de dados

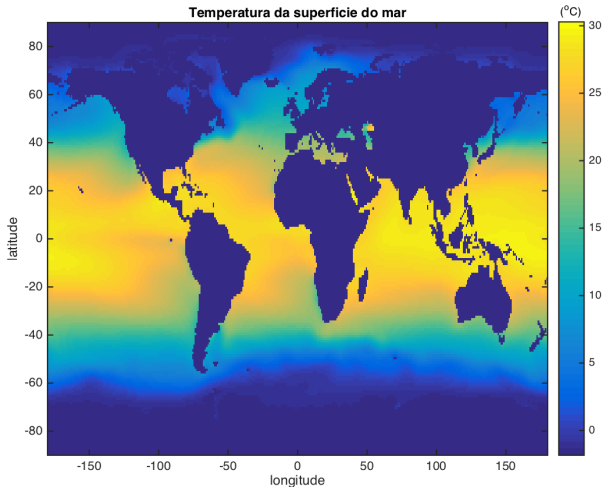
- NETCDF

World Ocean Atlas 2018

Conjunto de dados de climatologia de temperatura, salinidade, oxigênio e nutrientes

- <http://www.nodc.noaa.gov/>
- Escolher Access Data, World Ocean Atlas, temperatura
- Escolher: NETCDF, 1° de resolução espacial, "Objectively analyzed mean"
- Baixar o arquivo da média anual: "t00_01.nc"
- Usar o programa readnc_woa.m para ler o arquivo.

WOA18 - TSM



Faça mapas da temperatura para 200 m e 1000 m.