

SISTEMAS COMPUTACIONAIS E SEGURANÇA

AGOSTO

17/08 – Início das Aulas
24/08 – Aula 1
29 a 31/08 – **TECHWEEK**

SETEMBRO

07/09 – **FERIADO**
14/09 – Aula 2
21/09 – Aula 3
28/09 – Aula 4

OUTUBRO

05/10 – Aula 5
12/10 – **FERIADO**
19 e 20/10 – AVALIAÇÃO A1
26/10 – Aula 6

NOVEMBRO

02/11 – **FERIADO**
09/11 – Aula 7
16/11 – Aula 8
23/11 – Aula 9
30/11 – Aula 10

DEZEMBRO

04 a 08/12 – AVALIAÇÃO A3
11 e 12/12 – AVALIAÇÃO A2
14/12 – Término do semestre letivo

AVALIAÇÕES

A1 – Avaliação (30%)
A2 – Avaliação (30%)
A3 – Avaliação (40%)

Introdução



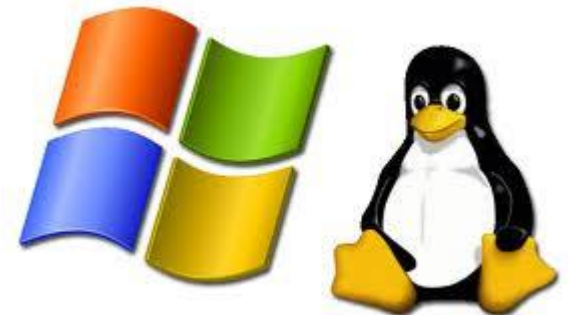
Sistemas Computacionais e Segurança

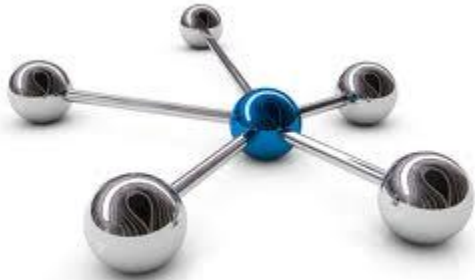


As empresas exigem ampla variedade de equipamentos tecnológicos, softwares e recursos de comunicação para funcionar e resolver problemas organizacionais. Sendo assim, você precisará de tecnologia e, como verá, existe uma grande variedade de opções, desde um simples desktop e/ou laptops até recentes lançamentos. Por exemplo: Drones, Robôs...



Você também precisará de software, porque cada computador exigirá uma sistema operacional e uma ampla gama de softwares aplicativos capazes de lidar com planilhas, documentos, imagens, arquivos de dados e outros.





Também precisaremos de várias redes, um local para os funcionários presentes fisicamente no escritório, e talvez acesso remoto, de maneira que os funcionários possam trocar arquivos e e-mails mesmo estando fora do escritório.

Você já teve curiosidade de saber como é um computador por dentro e como ele trabalha?

Quais são seus componentes e como eles se relacionam?



- ❑ Um computador precisa de vários circuitos interligados, componentes elétricos e eletrônicos para desempenhar suas funções. A parte física, que corresponde ao hardware, é comandada por conjuntos de instruções, conhecidos como programas, que definem os comandos que o computador deve executar. Assim, os programas correspondem ao software, a parte lógica do computador.
- ❑ Os computadores modernos são capazes de executar várias instruções básicas em frações de segundo e fazem muitos cálculos e comparações de modo bastante rápido. Diferentemente dos seres humanos, que usam o sistema decimal para fazer cálculos, os computadores utilizam o sistema binário, que lhes permite realizar suas operações de forma mais adequada.

Quanto aos computadores móveis e às redes sem fio, você já pensou e verificou como eles estão evoluindo rapidamente?



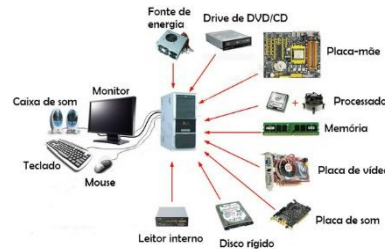
- ❑ Da telefonia celular 1G chegamos ao 5G, com possibilidades de interligação de objetos com a internet, ou seja, celular, televisão, relógios, óculos, canetas, câmeras de vídeo, janelas, lâmpadas e vários tipos de tecnologias que se interconectam à IoT (Internet of Things). A Internet das Coisas já está revolucionando a vida das pessoas com todos os tipos de computadores, possibilitando o acesso aos sistemas em qualquer hora e lugar, de modo transparente.
- ❑ Para suportar tanta tecnologia e inovação, é necessário aprofundar o conhecimento sobre segurança da informação, uma área da tecnologia da informação que trata da proteção de dados, pessoas, computadores e sistemas, garantindo a confiabilidade, integridade e disponibilidade dos sistemas computacionais, por meio de regras, normas, políticas de segurança, criptografia, controle de acesso e proteção da informação, por intermédio de hardware, software e algoritmos.

Conceitos Básicos



Um sistema de computação compreende dois elementos básicos:

- ❑ **Hardware:** conjunto de componentes mecânicos, elétricos e eletrônicos com os quais são construídos os computadores e equipamentos periféricos;

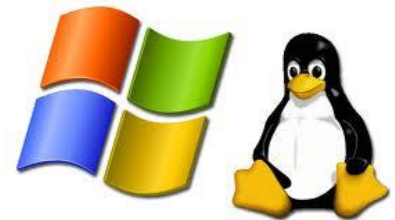


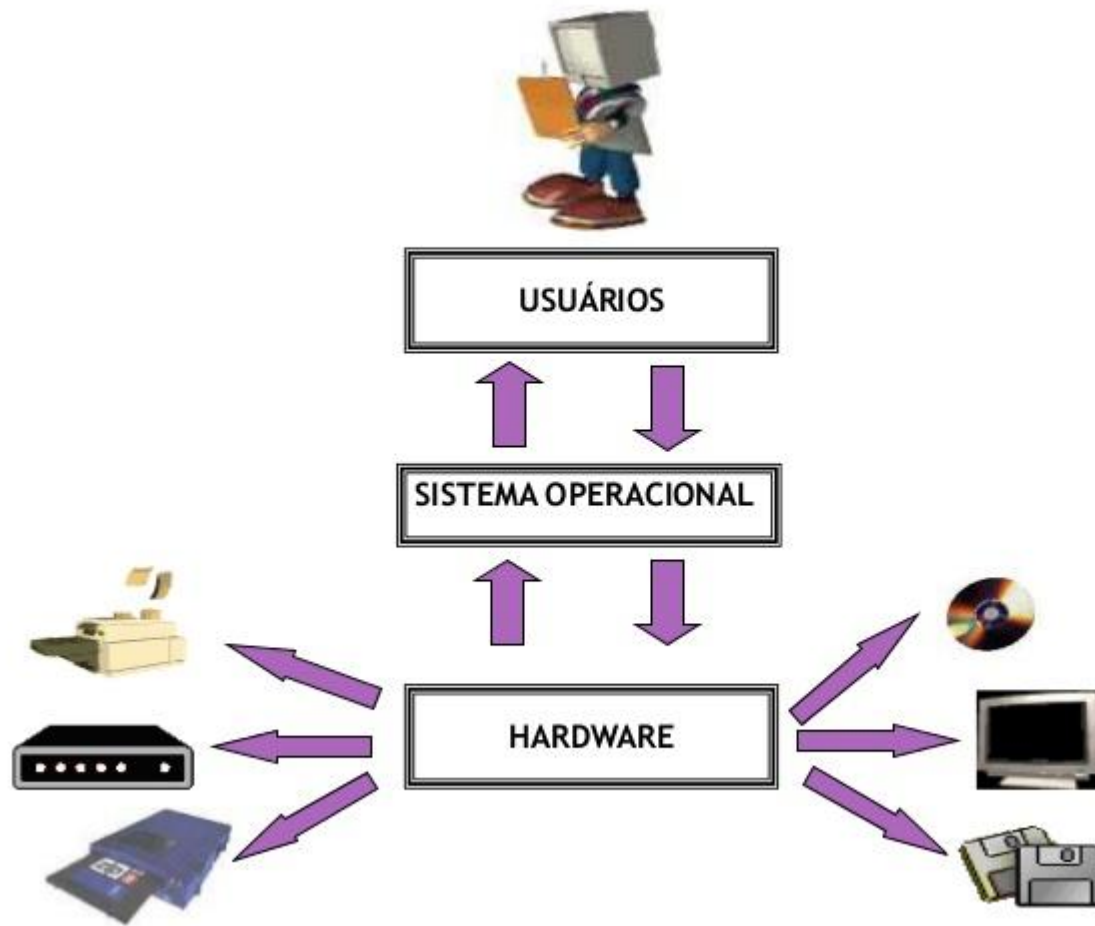
- ❑ **Software:** conjunto de programas, procedimentos e documentação que permitem utilizar o processamento fornecida pelo hardware.

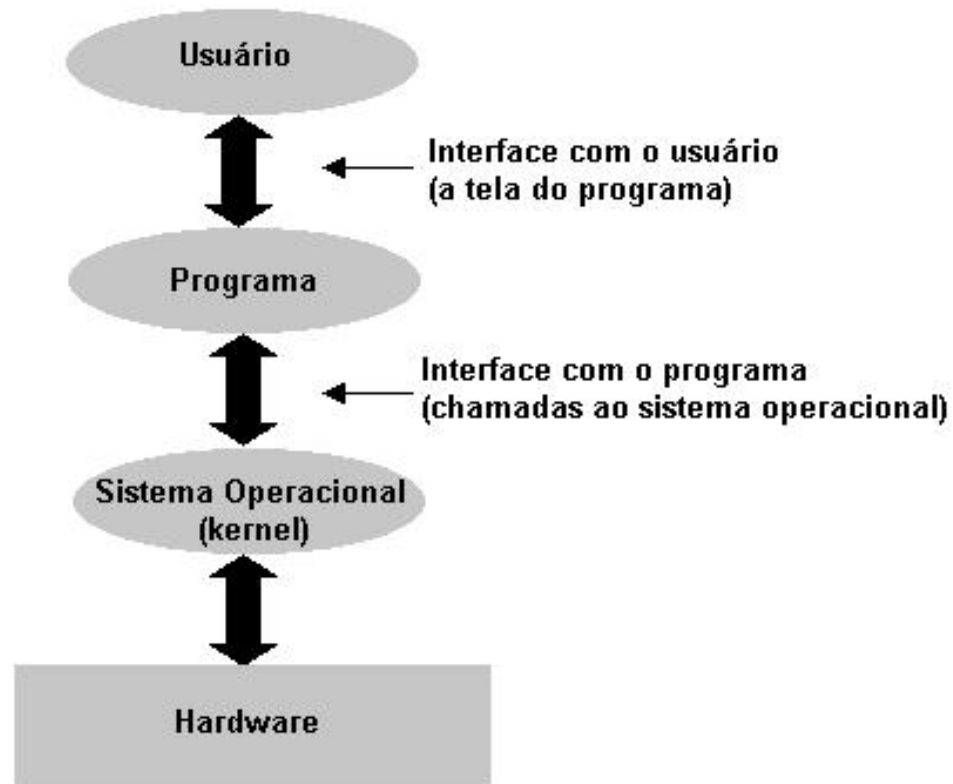


Software de sistema operacional

- ❑ **Gerencia e controla as atividades do computador;**
- ❑ **É composto de programas tradutores de linguagens computacionais que convertem as linguagens de programação em linguagem de máquina compreensível pelo computador;**
- ❑ **É o gerente-geral do sistema de computador.**







```
MikeOS CLI version 1.0
Inbuilt commands: DIR, CLS, HELP, PROMPT, VER, EXIT
> dir
MIKEKERN.BIN
ULAK.BIN
DOSTEST.BIN
GFXDEMO.BIN
HARDLIST.BIN
KEYBOARD.BIN
HELLO.BIN
> ver
MikeOS 0.30
>
>
>
> prompt
Enter a new prompt:
[mike@mikeos /]#
[mike@mikeos /]# hello
Hello, world! Press a key to exit...
[mike@mikeos /]#
[mike@mikeos /]# Woohoo!█
```



```
Terminal - glances
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
Ubuntu 14.04 64bit with Linux 3.13.0-32-generic on skywalker

CPU      3.3%      Carg  4-core  Mem    21.5%  active:  3.41G
user:    2.2%    nice:    0.0%  1 min:  0.33  total: 11.4G  inactive: 1.94G
system:  1.1%    iowait:  0.0%  5 min:  0.42  used:  2.46G  buffers:  359M
idle:    96.7%  irq:    0.0% 15 min:  0.52  free:  8.98G  cached:  3.11G

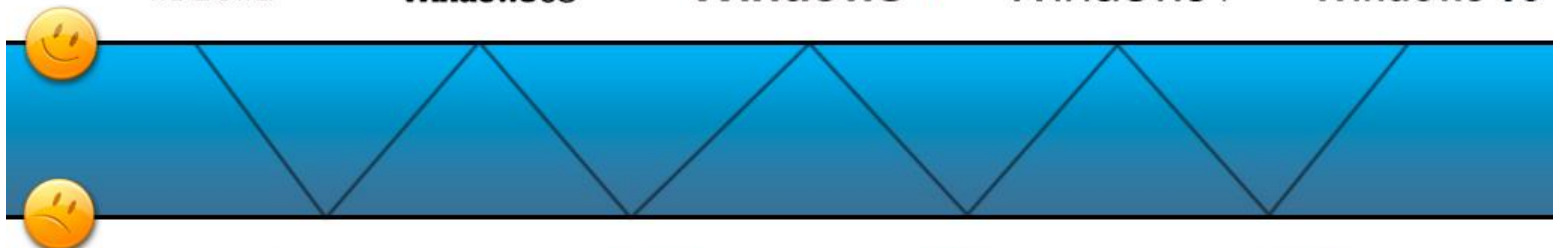
Network  Rx/s  Tx/s  Processes 244, 1 running, 243 sleeping, 0 other sorted by memory_percent
eth0     0b    0b
lo       0b    0b
wlan0    18Kb  5Kb

I/O do D  In/s  Out/s
sda1     0      0
sda2     0      0
sda3     0      0
sda4     0      0
sdb1     0      0

Parti    Usado  Total
/        6.59G 21.9G
oot/efi  3.69M 41.3M
/home    311G 431G
e/elcio  311G 431G
/run     1.39M 1.14G
systemd  0      0
/var     6.05G 23.3G

VIRT  RES  CPU%  MEM%  PID  USER      NI  S  TIME+  NAME
1.1G 360M  1.3   3.1 32196 elcio      0  S  0:19.36 chrome
1.1G 311M  0.0   2.7 32300 elcio      0  S 13:09.59 chrome
1.9G 294M  1.3   2.5 3193  elcio      0  S 27:37.10 /opt/google/chrome/chrome
918M 165M  1.6   1.4 15928 elcio      0  S  1:43.13 chrome
863M 159M  0.3   1.4 32303 elcio      0  S  0:14.76 chrome
904M 150M  1.0   1.3 16918 elcio      0  S  1:52.69 chrome
2.3G 134M  0.0   1.1 28766 elcio      0  S  0:55.87 dropbox
819M 127M  0.3   1.1 3263  elcio      0  S  5:09.70 chrome
670M 118M  0.3   1.0 1368  mysql      0  S  0:21.15 mysqld
826M  99M  0.0   0.8 13428 elcio      0  S  0:17.59 chrome
286M  49M  0.6   0.4 1400  root       0  S 10:04.40 Xorg
757M  47M  0.0   0.4 3266  elcio      0  S  0:14.87 chrome
863M  45M  0.0   0.4 2469  elcio      0  S  0:10.22 xfdesktop
751M  44M  0.0   0.4 3255  elcio      0  S  0:10.45 chrome
748M  43M  0.0   0.4 3249  elcio      0  S  0:03.56 chrome
766M  41M  0.0   0.4 3274  elcio      0  S  0:03.49 chrome
747M  41M  0.0   0.3 3299  elcio      0  S  0:01.70 chrome
745M  39M  1.0   0.3 3270  elcio      0  S  3:35.88 chrome
747M  38M  0.0   0.3 3282  elcio      0  S  0:01.66 chrome

Press 'h' for help                                     26-12-2014 16:57:37
```





```

1 - Default
File Edit Transfer Fonts Options Macro View Window Help
[Icons] [F] [F] [F] [Icons] [PA1] [PA2] [PA3] [ENT] [CLR]

SIGNON          UNIVERSITY OF MARYLAND AT COLLEGE PARK    DATE: 07/08/99
SYSTEM: ACCPRD1  ACADEMIC ADMINISTRATIVE INFORMATION SYSTEMS  TIME: 13:40:33
TERMINID: 0420   P R O D U C T I O N    C I C S

=====
Customer Assistance and Problem Reporting, call the Help desk at 301-405-7763.

          CCCCCC      IIIII      CCCCCC      SSSSSS
        CCCCCCCC      IIIII      CCCCCCCC      SSSSSSSS
      CCCC  CC      III      CCCC  CC      SSSS  SS
    CCC      III      CCC      SSSS
  CCC      III      CCC      SSSS
CCCC  CC      III      CCCC  CC      SS  SSSS
CCCCCCCC      IIIII      CCCCCCCC      SSSSSSSS
CCCCC      IIIII      CCCCCC      SSSSSS  4.1.0

Fill in your USERID and PASSWORD then press ENTER to sign on to CICS
  USERID:  _____  PASSWORD:          BYPASS INITIAL KEYWORD: _

PRESS: ENTER=Signon,   F1=Help,   F3=Exit CICS

1 Sess-1      129.2.128.7      21/13
  
```





Linguagem de programação de aplicativos

Para aplicações empresariais, as mais importantes linguagens de programação são, Java, C#, Cobol, Delphi, Php, .Net, ...



► Podcast

Definição de software e duas finalidades



Política de Cookies



Definição de software e duas finalidades

SOUNDCLOUD

Compartilhar



2:56

► 25

Tendências de software



Hoje, existem muitas fontes para obtenção de software e muito mais recursos para que os usuários criem suas próprias aplicações de software customizadas;

O uso crescente de software de código aberto e de ferramentas de software baseadas na nuvem exemplifica essa tendência.



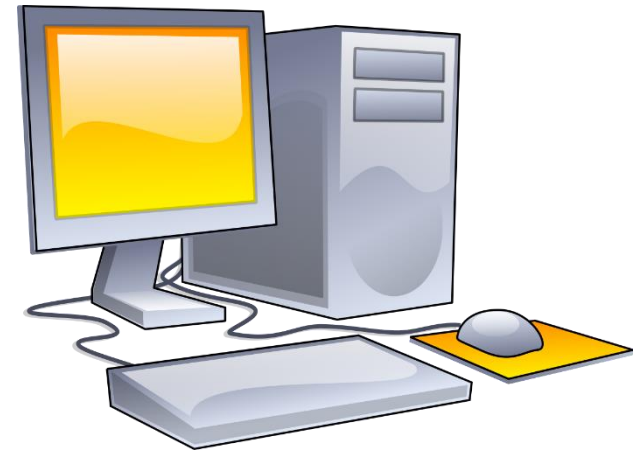
Benefícios trazidos pelo computador

- ✓ Confiabilidade e Exatidão
- ✓ Precisão no controle de processos
- ✓ Aumento da produtividade
- ✓ Análise de grandes quantidades de informação
- ✓ Auxílio à tomada de decisões
- ✓ Agilidade nas operações
- ✓ Redução da burocracia



Características do Computador

- ✓ Alta velocidade de processamento
- ✓ Alta capacidade de armazenamento
- ✓ Possibilidade de replicação
- ✓ Processamento ininterrupto
- ✓ Programável



O que é um Computador?



O que é um Computador?



O que é um Computador?



O que é um Computador?



Tipos de computadores

- Computadores pessoais (*desktop*)
- Computadores portáteis (*notebooks*)
- Computadores muito portáteis
- Servidores
- Mainframes
- Supercomputadores

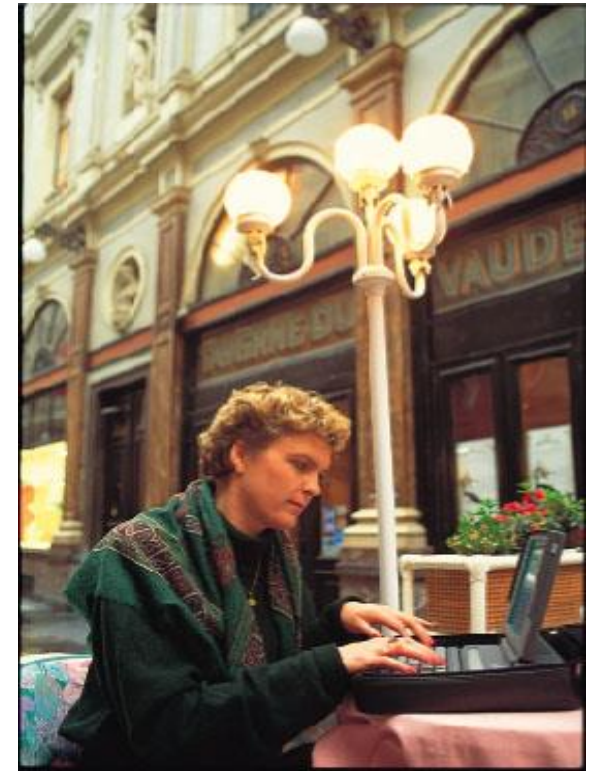


- Computadores de Mesa (desktop):
 - Também conhecidos como PC – *Personal Computer*, microcomputadores, ou computadores domésticos.

- Terminal de Rede:
 - Unidade central de processamento - CPU e memória mínima.
 - Projetado para ser usado em uma rede de computadores.
 - Às vezes chamado de cliente magro (*thin client*).
 - Realiza a interface entre o usuário e um servidor.

Tipos de computadores

- Computadores pequenos e leves – *notebooks, netbooks*.
- Suas capacidades se comparam às dos computadores de mesa:
 - Processamento e memória similares.
 - Disco rígido, CD/DVD...
- Capacidade de conexão em rede cabeada e sem fio (Wi Fi).



Tipos de computadores

➤ Exemplos

- Palm
- Pocket PC
- BlackBerry
- Smartphones

➤ Usos

- Agenda de compromissos, contatos, tarefas.
- Rodam versões reduzidas de software: processador de texto, planilhas eletrônicas, editores de texto, e-mail, web...
- Acesso sem fio (Wi-Fi ou celular) à Internet.



Mainframes

- Computadores muito grandes e potentes:
 - Capazes de processar bilhões de instruções por segundo.
 - Grande capacidade de armazenamento de dados
- Frequentemente usados para aplicações com milhares de usuários:
 - Sistemas de reservas de passagens aéreas, hotéis
 - Sistemas financeiros (bancos, seguradoras, financeiras)
 - Servidores de bancos de dados corporativos

Mainframes



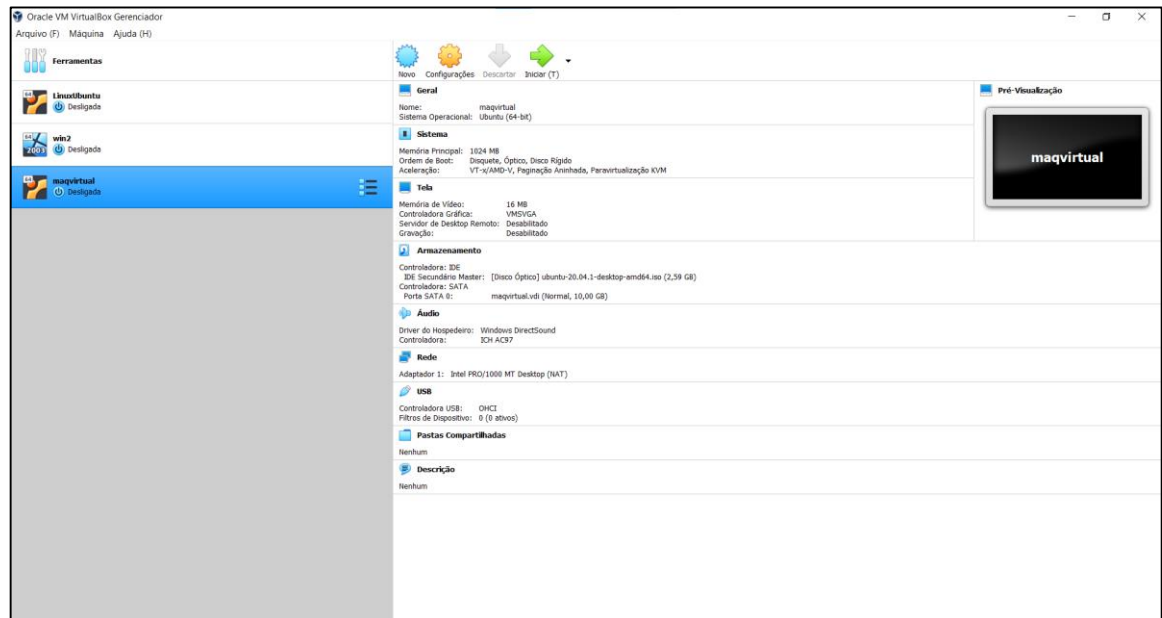
Supercomputadores

- Os computadores mais rápidos e mais poderosos:
 - Capazes de processar trilhões de instruções por segundo.
- Usados para aplicações muito sofisticadas que requerem gigantescas manipulações de dados:
 - Previsão do tempo.
 - Simulações e cálculos de alta precisão.
 - Efeitos especiais para cinema.
 - Militares.



GERANDO UMA MÁQUINA VIRTUAL

- Baixe o Virtual Box em: <https://www.virtualbox.org/>
- Baixe uma versão linux (cuidado com o porte de sua máquina) ou outro Sistema Operacional de sua escolha. <https://www.ubuntu.com/download/desktop>



<https://bellard.org/jslinux/>

JSLinux

Run Linux or other Operating Systems in your browser!

The following emulated systems are available:

CPU	OS	User Interface	VFsync access	Startup Link	TEMU Config	Comment
x86	Alpine Linux 3.12.0	Console	Yes	click here	url	
x86	Alpine Linux 3.12.0	X Window	Yes	click here	url	Right mouse button for the menu.
x86	Buildroot (Linux)	Console	Yes	click here	url	
x86	Buildroot (Linux)	X Window	Yes	click here	url	Right mouse button for the menu.
x86	Windows 2000	Graphical	No	click here	url	Disclaimer.
x86	FreeDOS	VGA Text	No	click here	url	
riscv64	Buildroot (Linux)	Console	Yes	click here	url	
riscv64	Buildroot (Linux)	X Window	Yes	click here	url	Right mouse button for the menu.
riscv64	Fedora 29 (Linux)	Console	Yes	click here	url	Warning: longer boot time.
riscv64	Fedora 29 (Linux)	X Window	Yes	click here	url	Warning: longer boot time. Right mouse button for

```

Loading...
Welcome to JS/Linux (i586)
Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# pwd
/root
localhost:~# █
  
```



```
Loading...
Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.
```

```
localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# pwd
/root
localhost:~#
```

Comandos básicos:

- | | |
|----------|--------------|
| - Pwd | - Diff |
| - Cd | - Tar |
| - Ls | - Chmod |
| - Cat | - Chown |
| - Cp | - Jobs |
| - Mv | - Kill |
| - Mkdir | - Ping |
| - Rmdir | - Wget |
| - Rm | - Uname |
| - Touch | - Top |
| - Locate | - History |
| - Find | - Man |
| - Grep | - Echo |
| - Sudo | - Zip, unzip |
| - Df | - Hostname |
| - Du | - Useradd, |
| - Head | userdel |
| - Tail | |

/ raiz (root) é o diretório de nível mais alto. Contém todos os demais subdiretórios. Esse diretório é a referência para o caminho dos demais diretório (path) (serve para o caminho absoluto – onde o **caminho absoluto** sempre usa esse diretório como origem)

/bin: contém os arqs binários- utilitários – comandos dos usuários comuns

/boot: contém arqs de inicialização do SO

/dev: contém arqs que mapeiam os dispositivos (hd, mouse,)

/etc: contém arqs de configuração do SO (passwd – contém as contas de usuários,...)

/home: contém os diretórios dos usuários do SO

/lib: bibliotecas do SO (auxiliam à confecção de programas,...)

/media: diretório que serve para montar dispositivos removíveis (cdrom, pendrive,...)

/mnt: em geral, usado para montar partições de outros SOs

Obs: o SO Linux usa o sistema de arquivos ext2, ext3, ext4 – o Windows usa sistema de arquivos NTFS (fat32). Esses sistemas de arquivos costumam estar associados às formatações dos discos e suas partições)

/opt: em geral, serve para instalar programas (sistemas) proprietários

/proc: diretório virtual - NÃO é um diretório propriamente dito (não possui espaço no disco) Ele apenas reside em Memória. Nele vemos os números dos processos (PID) e informações desses processos. Além disso, pode-se ver informações do hardware (exe: cpuinfo => mostra informações da CPU,...

/root: trata-se do diretório do administrador do SO – chamado de root

/run: apresenta informações dos processos em tempo de execução

/sbin: contém arquivos binários (parece com o /bin) com utilitários (comandos) do administrador do SO – não são comandos comuns (ex: fdisk – usado pelo administrador-serve para gerenciar o disco -formatar o disco...)

/srv: em geral, contém sistemas open source com serviços

/sys: é similar ao /proc, portanto o /sys é um diretório virtual. Contém informações similares ao /proc só que com organização diferente

/tmp: diretório temporário, em geral, tem os seus dados apagados quando se reinicia o SO. Esse diretório pode ter acesso de leitura e gravação de vários usuários, portanto, é necessário ter certo cuidado ao deixar um arq. nele, pois outro usuário pode, por exemplo, apagar o arquivo.

Obs: esse diretório não é tão protegido quanto os demais. Isso significa que outros usuários podem apagar ou incluir arquivos no tmp. Em geral, os arquivos contidos em tmp são apagados ao se reinicializar

/usr: diretório que, em geral, contém configuração dos ambientes dos usuários cadastrados no SO

/var: contém arqs variáveis. Nele existe os arqs que serão impressos e, depois disso, apagados.

Contém arquivos de mensagens (mail) e o **importante subdiretório 'log' que contém o registro de ocorrências de hardware, software e dos usuários.**

Obs: diretório que contém o spool de impressão (arqs que aguardam na fila para serem impressos), mail possui mensagens e tem o importante subdiretório chamado de log. O log contém informações de ocorrências tanto de hardware, quanto dos usuários- esse diretório é de fundamental importância para a área de segurança.

FHS : Filesystem Hierarchy Standard

a) Mostrar o comando para indicar o seu diretório atual (mostra o path do diretório onde você está)

R: pwd

b) Mostrar o conteúdo do diretório que contém os dispositivos do sistema operacional

R: ls -l /dev

c) Mostrar o conteúdo do diretório que contém os arquivos de configuração do sistema operacional

R: ls -l /etc

d) Mostrar o conteúdo do diretório do administrador do sistema operacional

R: ls -l /root

e) Mostrar o conteúdo do diretório que contém os arquivos temporários do sistema operacional

R: ls -l /tmp

f) Mostrar o conteúdo do diretório raiz do sistema operacional

R: ls -l /

g) Considere o FHS. Apresentar os diretórios do FHS e os seus conteúdos.

R: ls /

h) Apresentar os comandos básicos e descrever suas funcionalidades.

Vide notas de aula

Vi oi.sh

```
#!/bin/bash  
echo "Oi, tudo bem?"  
  
~  
~
```

Digitar os comandos “:wq” → Enter

```
localhost:/professor# chmod a+x Oi.sh  
localhost:/professor# ls  
Oi.sh  
localhost:/professor# ls -l  
total 4  
-rwxr-xr-x  1 root    root      34 Aug 24 22:28 Oi.sh  
localhost:/professor# ./Oi.sh  
Oi, tudo bem?
```


- Mostrar o **CPU** virtual que está sendo simulado pelo JSLinux com **1scpu**

```
localhost:~#  
localhost:~#  
localhost:~# lscpu  
Architecture:          i586  
CPU op-mode(s):        32-bit  
Byte Order:            Little Endian  
Address sizes:         32 bits physical, 32 bits virtual  
CPU(s):                1  
On-line CPU(s) list:   0  
Thread(s) per core:    1  
Core(s) per socket:    1  
Socket(s):             1  
Vendor ID:             AuthenticX86  
CPU family:            5  
Model:                 4  
Model name:            05/04  
Stepping:              3  
CPU MHz:               100.000  
BogoMIPS:              200.00  
Virtualization:        AMD-V  
Flags:                 fpu tsc msr pae cx8 cmov clflush mmx fxsr sse sse2 rdtscp cpuid svm npt  
localhost:~#
```

- Mostrar a **quantidade de memória** do sistema com **free**
 - Usar as opções **free -b** para mostrar em bytes, **free -k** em kiB, **free -m** em MiB

```
localhost:~#
localhost:~# free -b
              total        used        free      shared  buff/cache   available
Mem:      191737856    5021696    185765888        4096     950272    183549952
Swap:           0           0           0
localhost:~# free -k
              total        used        free      shared  buff/cache   available
Mem:       187244        4904       181412         4         928       179248
Swap:           0           0           0
localhost:~# free -m
              total        used        free      shared  buff/cache   available
Mem:         182           4         177         0           0         175
Swap:          0           0           0
localhost:~#
```

- Compilar um programa em C usando o `gcc` e executá-lo

```
localhost:~#  
localhost:~# cat > programa.c  
#include <stdio.h>  
  
float a, b, c;  
  
int main() {  
    a = 3.14;  
    b = 5.2;  
    c = a * b + 1;  
    printf("%f\n", c);  
    return 0;  
}  
localhost:~# gcc programa.c -o programa  
localhost:~# ./programa  
17.327999  
localhost:~#
```

Editores Linux

Vi programa.c

Editar = i

Sair e gravar = <ESC> :wq

OU

Nano programa.c

Isso vai gerar um arquivo executável (opção -o). Usar control-D para terminar o comando `cat`.

Ao executar, o resultado da operação é exibido no terminal.

- Mostrar o conteúdo do arquivo executável gerado (arquivo "programa") em hexadecimal usando `xxd`

```
localhost:~# xxd programa
00000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF.....
00000010: 0300 0300 0100 0000 6810 0000 3400 0000  .....h...4...
00000020: 7c42 0000 0000 0000 3400 2000 0a00 2800  |B.....4. ...(.
00000030: 2100 2000 0600 0000 3400 0000 3400 0000  !. ....4...4...
00000040: 3400 0000 4001 0000 4001 0000 0400 0000  4...@...@.....
00000050: 0400 0000 0300 0000 7401 0000 7401 0000  .....t...t...
00000060: 7401 0000 1700 0000 1700 0000 0400 0000  t.....
00000070: 0100 0000 0100 0000 0000 0000 0000 0000  .....
00000080: 0000 0000 6003 0000 6003 0000 0400 0000  .... ..
00000090: 0010 0000 0100 0000 0010 0000 0010 0000  .....
000000a0: 0010 0000 cd02 0000 cd02 0000 0500 0000  .....
000000b0: 0010 0000 0100 0000 0020 0000 0020 0000  .....
000000c0: 0020 0000 b800 0000 b800 0000 0400 0000  . ....
000000d0: 0010 0000 0100 0000 fc2e 0000 fc3e 0000  .....>..
000000e0: fc3e 0000 0801 0000 3401 0000 0600 0000  .>.....4.....
000000f0: 0010 0000 0200 0000 0c2f 0000 0c3f 0000  ...../...?..
00000100: 0c3f 0000 c000 0000 c000 0000 0600 0000  .?.....
00000110: 0400 0000 50e5 7464 0c20 0000 0c20 0000  ....P.td. ...
00000120: 0c20 0000 2400 0000 2400 0000 0400 0000  . ..$...$.....
00000130: 0400 0000 51e5 7464 0000 0000 0000 0000  ....Q.td.....
00000140: 0000 0000 0000 0000 0000 0000 0600 0000  .....
00000150: 1000 0000 52e5 7464 fc2e 0000 fc3e 0000  ....R.td.....>..
00000160: fc3e 0000 0401 0000 0401 0000 0400 0000  .>.....
```

- Mostrar o arquivo executável decodificado em instruções de máquina usando

```
objdump -d
```

```
localhost:~#
localhost:~# objdump -d programa

programa:      file format elf32-i386

Disassembly of section .init:

00001000 <_init>:
 1000:      83 ec 0c                sub    $0xc,%esp
 1003:      e8 be 01 00 00        call   11c6 <frame_dummy>
 1008:      e8 8d 02 00 00        call   129a <__do_global_ctors_aux>
 100d:      83 c4 0c                add    $0xc,%esp
 1010:      c3                    ret

Disassembly of section .plt:

00001020 <.plt>:
 1020:      ff b3 04 00 00 00      pushl  0x4(%ebx)
 1026:      ff a3 08 00 00 00      jmp     *0x8(%ebx)
 102c:      00 00                  add     %al,(%eax)
  ...

00001030 <printf@plt>:
 1030:      ff a3 0c 00 00 00      jmp     *0xc(%ebx)
 1036:      68 00 00 00 00        push    $0x0
 103b:      e9 e0 ff ff ff        jmp     1020 <.plt>

00001040 <__libc_start_main@plt>:
 1040:      ff a3 10 00 00 00      jmp     *0x10(%ebx)
 1046:      68 08 00 00 00        push    $0x8
 104b:      e9 d0 ff ff ff        jmp     1020 <.plt>

Disassembly of section .plt.got:
```

- Mostrar no código desassemblado as instruções que fazem os cálculos que aparecem no programa fonte (multiplicação e adição de ponto flutuante). Mostrar que as instruções são executadas em uma sequência de endereços de memória crescentes.

```

1210:      c3                ret
00001211 <__x86.get_pc_thunk.cx>:
1211:      8b 0c 24          mov     (%esp), %ecx
1214:      c3                ret
00001215 <main>:
1215:      8d 4c 24 04       lea     0x4(%esp), %ecx
1219:      83 e4 f0          and     $0xffffffff0, %esp
121c:      ff 71 fc          pushl  -0x4(%ecx)
121f:      55                push   %ebp
1220:      89 e5             mov     %esp, %ebp
1222:      53                push   %ebx
1223:      51                push   %ecx
1224:      e8 e4 ff ff ff    call    120d <__x86.get_pc_thunk.ax>
1229:      05 a3 2d 00 00     add     $0x2da3, %eax
122e:      8d 90 60 00 00 00   lea     0x60(%eax), %edx
1234:      d9 80 38 e0 ff ff   flds    -0x1fc8(%eax)
123a:      d9 1a             fstps   (%edx)
123c:      8d 90 58 00 00 00   lea     0x58(%eax), %edx
1242:      d9 80 3c e0 ff ff   flds    -0x1fc4(%eax)
1248:      d9 1a             fstps   (%edx)
124a:      8d 90 60 00 00 00   lea     0x60(%eax), %edx
1250:      d9 02             flds    (%edx)
1252:      8d 90 58 00 00 00   lea     0x58(%eax), %edx
1258:      d9 02             flds    (%edx)
125a:      de c9             fmulp   %st, %st(1)
125c:      d9 e8             rtd     1
125e:      de c1             faddp   %st, %st(1)
1260:      8d 90 5c 00 00 00   lea     0x5c(%eax), %edx
1266:      d9 1a             fstps   (%edx)
1268:      8d 90 5c 00 00 00   lea     0x5c(%eax), %edx
126e:      d9 02             flds    (%edx)
1270:      83 ec 04          sub     $0x4, %esp

```

Obrigado