

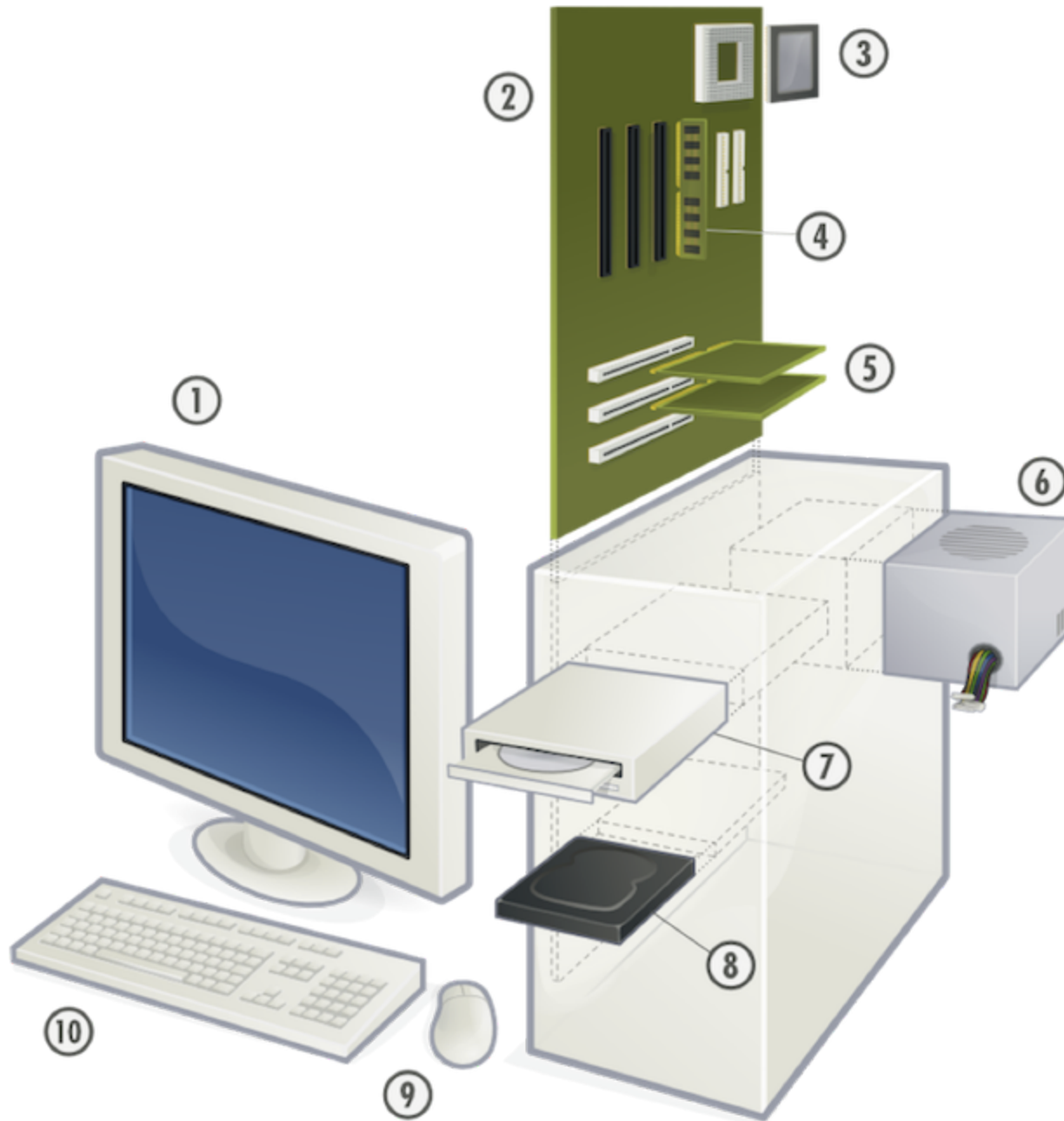
Introdução à Programação de Computadores para Biologia

Computadores e Introdução ao Unix

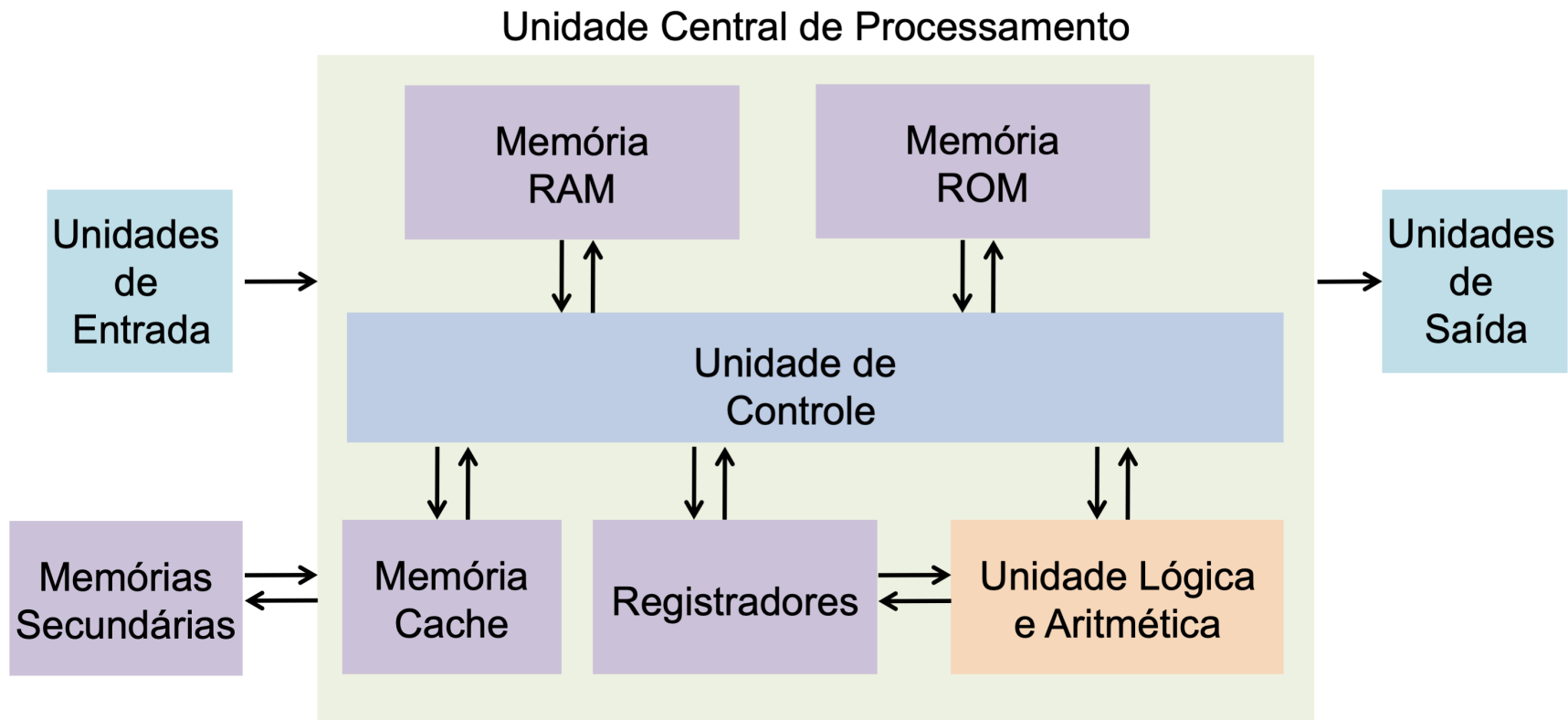
Aula 02

<https://tttorres.github.io/introprog2024/>

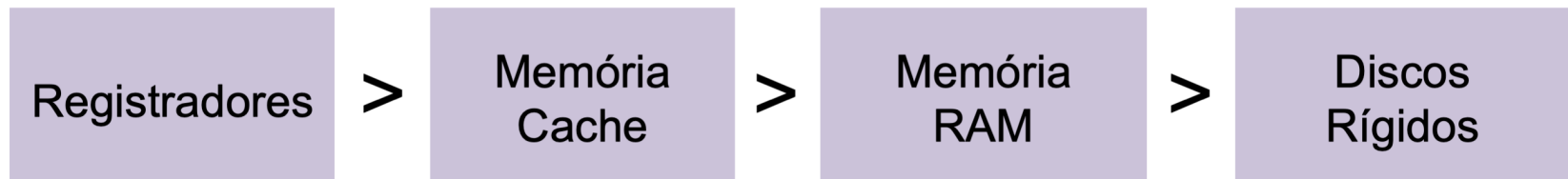
COMPUTADORES



HARDWARE: CPU



HARDWARE: Memórias



HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

"0" desligado

"1" ligado

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

"0" desligado

"1" ligado

Unidade de informação = *bit*

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

"0" desligado

"1" ligado

1 *bit* pode representar dois valores: 0 e 1

HARDWARE -> SOFTWARE

2 *bits* podem representar quatro valores:

bit1bit2	valor
00	0
01	1
10	2
11	3

HARDWARE -> SOFTWARE

3 *bits* podem representar oito valores:

bit1bit2bi3	valor
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

HARDWARE -> SOFTWARE

3 *bits* podem representar oito valores: 2^n

bit1bit2bi3	valor
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

"0" desligado

"1" ligado

Unidade de informação = *bit*

Conjunto de 8 *bits* = *byte*

(é, tradicionalmente, o número de bits para representar um caracter)

1 byte pode ter até $2^8 = 256$ configurações diferentes

HARDWARE -> SOFTWARE

NOME	Memória
bit	{0,1}
byte	8 bits
kilobyte	2^{10} bytes (1024)
megabyte	2^{20} bytes (pouco mais de 1 milhão de bytes)
gigabyte	2^{30} bytes (pouco mais de 1 bilhão de bytes)
terabyte	2^{40} bytes (pouco mais de 1 trilhão de bytes)

HARDWARE/SOFTWARE

Base decimal

526

$$526 = 5 * 10^2 + 2 * 10^1 + 6 * 10^0$$

$$526 = 500 + 20 + 6$$

HARDWARE/SOFTWARE

Base decimal

38

$$38 = 3 * 10^1 + 8 * 10^0$$

$$38 = 30 + 8$$

HARDWARE/SOFTWARE

Base decimal

38

$$38 = 3 * 10^1 + 8 * 10^0$$

$$38 = 30 + 8$$

Base binária

100110₂

HARDWARE/SOFTWARE

Base decimal

38

$$38 = 3 * 10^1 + 8 * 10^0$$

$$38 = 30 + 8$$

Base binária

100110₂

$$\begin{aligned} &1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = \\ &= 32 + 0 + 0 + 4 + 2 + 0 = 38 \end{aligned}$$

HARDWARE/SOFTWARE

Conversão de Base Decimal para Binária

Exemplo: 126

126 / 2 =	63	0
63 / 2 =	31	1
31 / 2 =	15	1
15 / 2 =	7	1
7 / 2 =	3	1
3 / 2 =	1	1
1 / 2 =		1



[illegible]

SOFTWARE

Linguagens de programação

- conjunto de convenções e regras que especificam como instruir o computador a executar determinadas tarefas
- serve como meio de comunicação entre o indivíduo que deseja resolver um determinado problema e o computador

SOFTWARE

Linguagens de programação

Classificação de linguagens

- linguagens em nível de máquina
- linguagens de montagem (Assembly)
- linguagens orientadas ao usuário (C, C++, Python, Perl)
- linguagens orientadas à aplicação (SQL, Matlab, S, R)

SOFTWARE

Linguagens em nível de máquina

instrução 0010 0001 0110 1100

realiza a soma (código de operação 0010) do dado armazenado no registrador 0001, com o dado armazenado na posição de memória 108 (0110 1100)

SOFTWARE

O programa mais escrito do mundo

```
01001000 01100101 01101100 01101100 01101111 00101100  
00100000 01010111 01101111 01110010 01101100 01100100  
00100001
```

SOFTWARE

O programa mais escrito do mundo

```
01001000 01100101 01101100 01101100 01101111 00101100  
00100000 01010111 01101111 01110010 01101100 01100100  
00100001
```

Imprimir **"Hello, world!"** na tela

HELLO WORLD

[illegible]

HELLO WORLD

```
section .text
    global _start                ;must be declared for linker

(ld)_start:                    ;tell linker entry point

    mov edx,len ;message length
    mov ecx,msg ;message to write
    mov ebx,1   ;file descriptor (stdout)
    mov eax,4   ;system call number (sys_write)
    int 0x80    ;call kernel

    mov eax,1   ;system call number (sys_exit)
    int 0x80    ;call kernel

section .data
msg db 'Hello, world!',0xa ;our dear string
len equ $ - msg           ;length of our dear string
```

Linux assembler

HELLO WORLD

C

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

BASIC

```
10 PRINT "Hello world!"
20 END
```

JAVA

```
public class HelloWorld
{
    public static void main(string[] args)
    {
        System.out.println("Hello world!");
    }
}
```

<http://helloworldcollection.de>

HELLO WORLD

Pascal

```
program HelloWorld;  
begin  
    writeln('Hello World!');  
end.
```

Python

```
print "Hello World!"
```

Perl

```
print "Hello World!";
```

<http://helloworldcollection.de>

SCRIPTS vs PROGRAMAS

C

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

Perl

```
print "Hello World!";
```

SCRIPTS vs PROGRAMAS

```
#!/usr/bin/perl

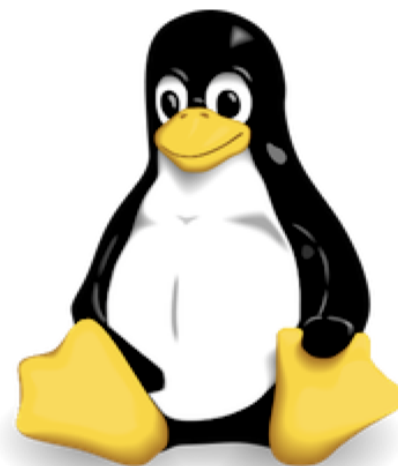
my ($E, $M, $H, $nota_final); #$E, easy; $M, medium; $H, hard
($E, $M, $H) = @ARGV;

$nota_final = ((2*$E)+(3*$M)+(5*$H))/10;

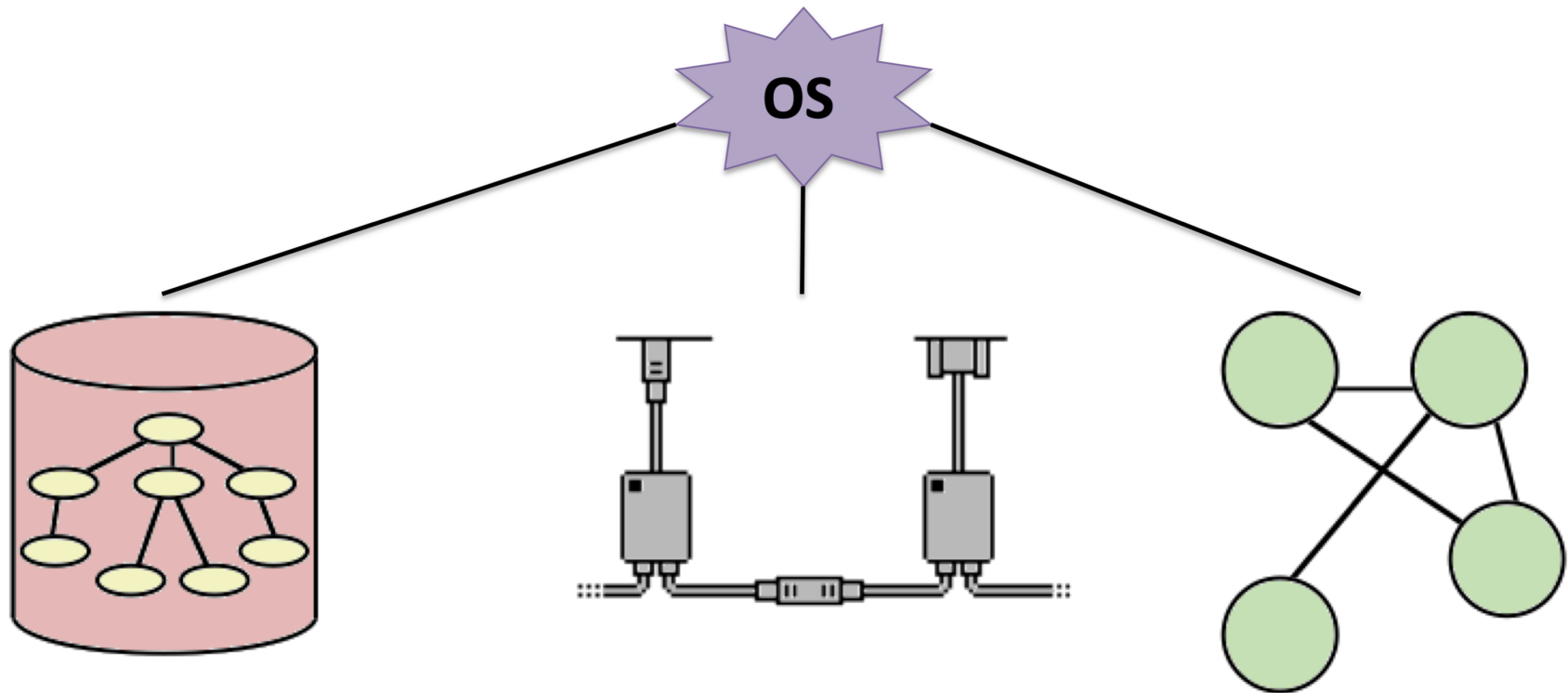
if ($nota_final >= 5) {
    print "Aluno aprovado\n";
} else {
    print "Aluno reprovado\n";
}

#conversao de nota para conceitos, apenas para a Pos-graduacao
if ($nota_final < 5) {
    print "R, Reprovado, sem direito a credito\n";
} elsif ($nota_final <= 7.0) {
    print "C, Regular, com direito a credito\n";
} elsif ($nota_final <= 8.5) {
    print "B, Bom, com direito a credito\n";
} else {
    print "A, Excelente, com direito a credito\n";
}
exit;
```

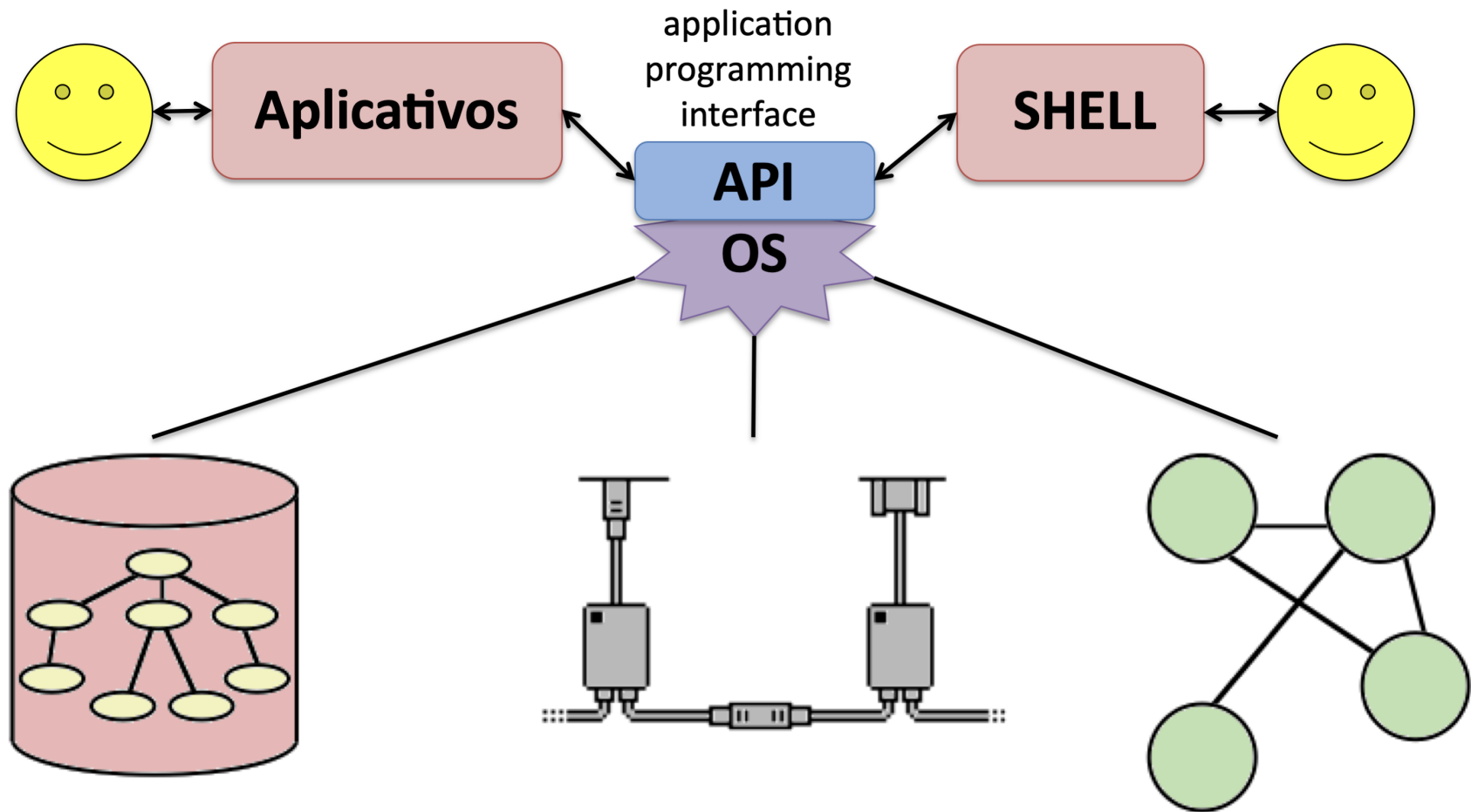
SISTEMA OPERACIONAL



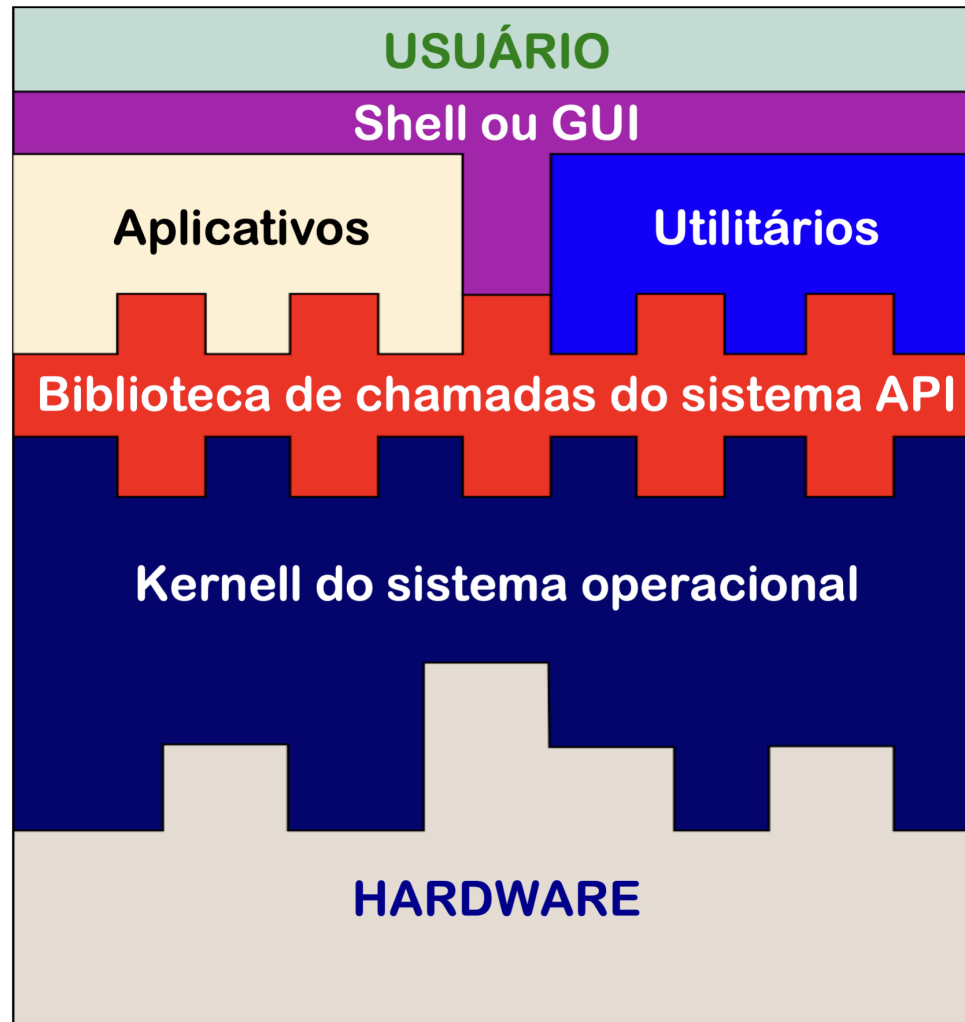
SISTEMA OPERACIONAL



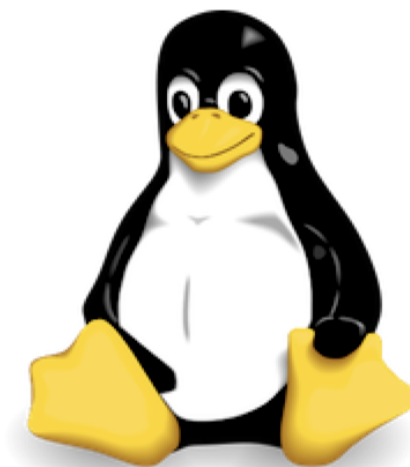
SISTEMA OPERACIONAL



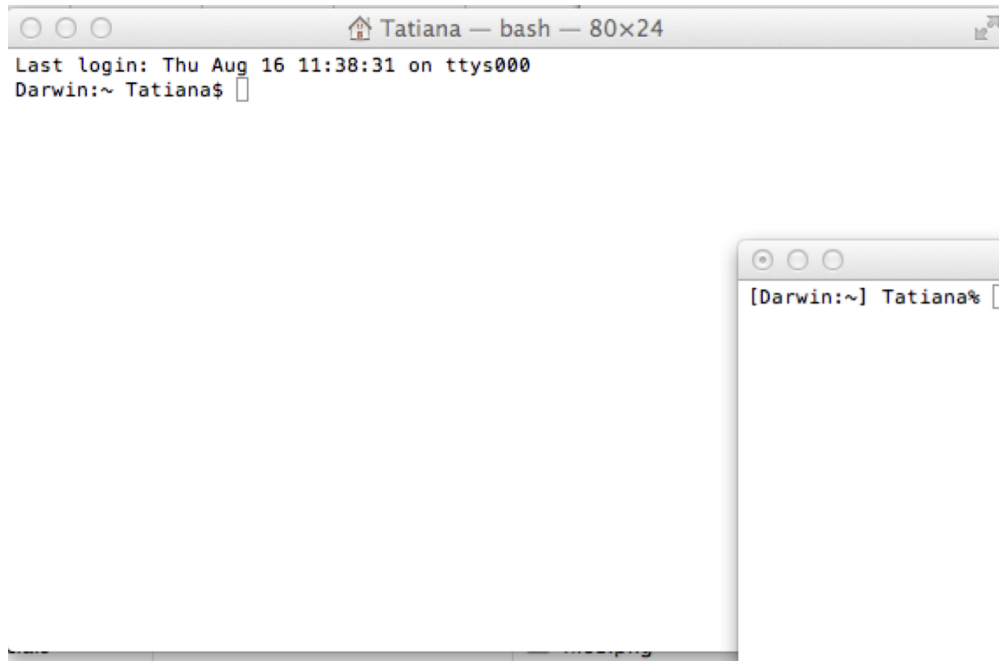
SISTEMA OPERACIONAL



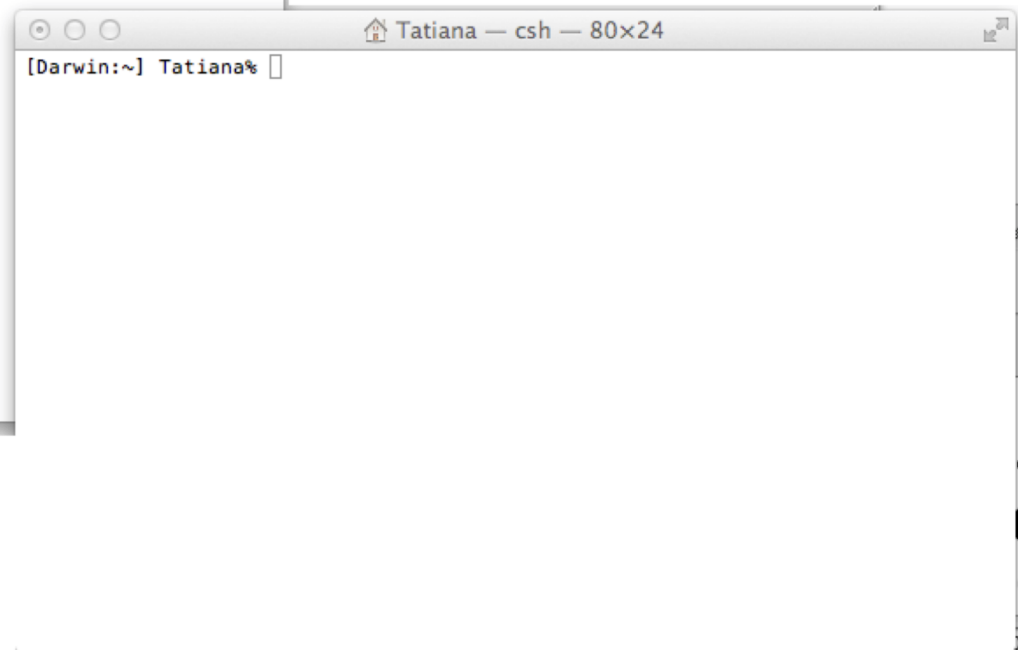
SISTEMA OPERACIONAL: UNIX



SISTEMA OPERACIONAL: UNIX SHELL



```
○ ○ ○ Tatiana — bash — 80x24
Last login: Thu Aug 16 11:38:31 on ttys000
Darwin:~ Tatiana$
```



```
○ ○ ○ Tatiana — csh — 80x24
[Darwin:~] Tatiana%
```

UNIX

Comandos mais utilizados

1. Comando `ls` (list): listando arquivos e diretórios

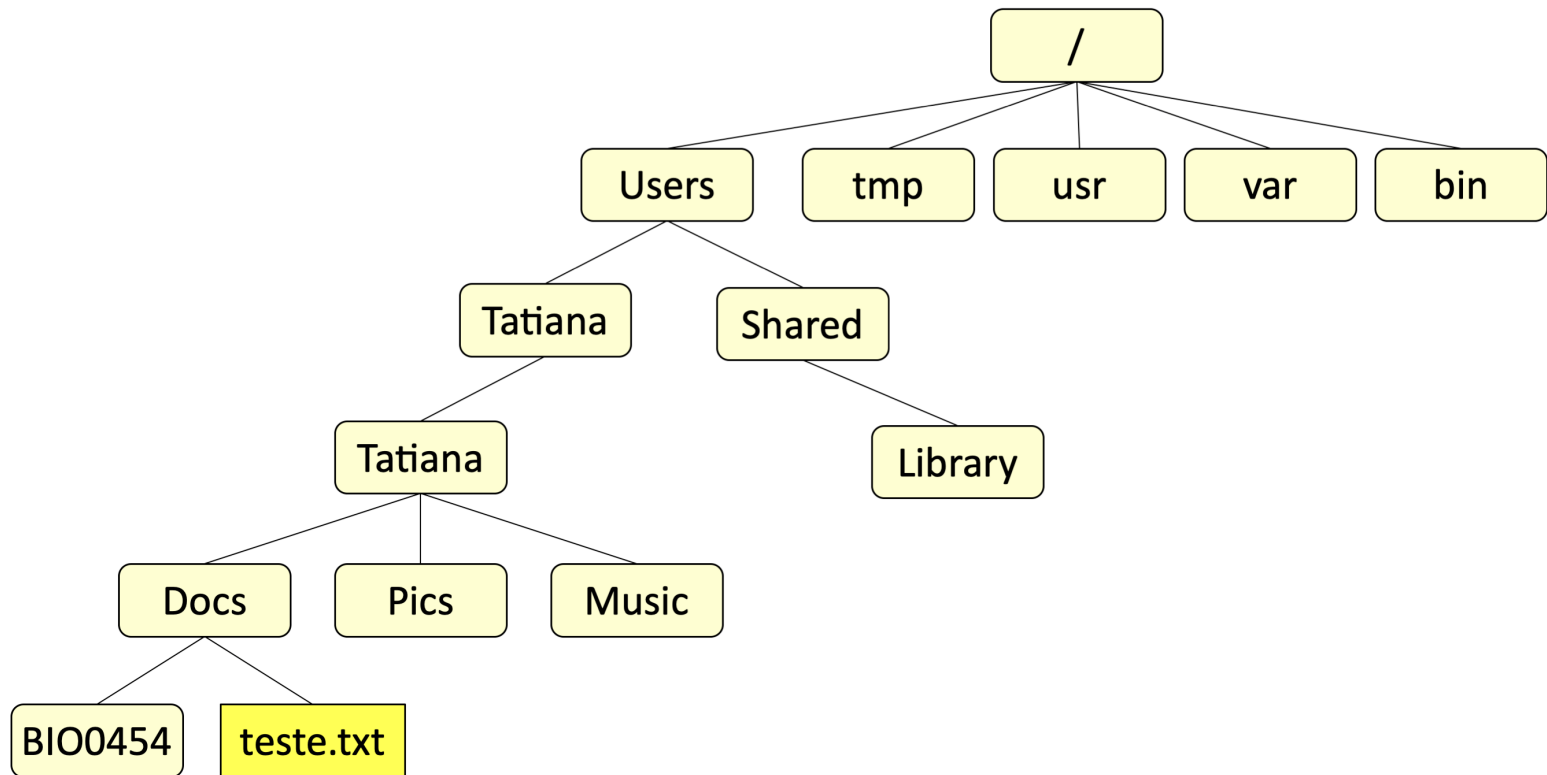
```
ls  
ls -a  
ls -l  
ls -la
```

OBS: *man* (manual)

UNIX

Comandos mais utilizados

2. "Navegando" em diretórios



UNIX

Comandos mais utilizados

2. "Navegando" em diretórios

a. Verificar o diretório atual

b. Ir para a raiz

c. Voltar para a home

UNIX

Comandos mais utilizados

2. "Navegando" em diretórios

a. Verificar o diretório atual: *pwd* (print working directory)

```
pwd
```

b. Ir para a raiz

c. Voltar para a home

UNIX

Comandos mais utilizados

2. "Navegando" em diretórios

a. Verificar o diretório atual: *pwd* (print working directory)

```
pwd
```

b. Ir para a raiz: *cd* (change directory)

```
cd /
```

c. Voltar para a home

UNIX

Comandos mais utilizados

2. "Navegando" em diretórios

a. Verificar o diretório atual: *pwd* (print working directory)

```
pwd
```

b. Ir para a raiz: *cd* (change directory)

```
cd /
```

c. Voltar para a home: *cd*

```
cd ~
```

UNIX

Comandos mais utilizados

PREPARAÇÃO

Na página da disciplina, baixar s seguinte arquivo na pasta "home":

- Arquivo em formato fasta: dmel-gene-r5.45.fasta

<https://tttorres.github.io/introprog2024/>

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

- mkdir (make directory)
- cp (copy)
- mv (move)
- rm (remove; MUITA CALMA!!!)

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

a. Criar a pasta myseq no home (mkdir)

b. Mover o arquivo dmel-gene.fasta para myseq (mv)

c. Ir para a pasta myseq (cd)

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

a. Criar a pasta myseq no home (mkdir)

```
mkdir myseq
```

b. Mover o arquivo dmel-gene.fasta para myseq (mv)

c. Ir para a pasta myseq (cd)

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

a. Criar a pasta myseq no home (mkdir)

```
mkdir myseq
```

b. Mover o arquivo dmel-gene.fasta para myseq (mv)

c. Ir para a pasta myseq (cd)

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

a. Criar a pasta myseq no home (mkdir)

```
mkdir myseq
```

b. Mover o arquivo dmel-gene.fasta para myseq (mv)

```
mv /mnt/c/Users/Aluno/dmel-gene.fasta ~/myseq/
```

c. Ir para a pasta myseq (cd)

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

a. Criar a pasta myseq no home (mkdir)

```
mkdir myseq
```

b. Mover o arquivo dmel-gene.fasta para myseq (mv)

```
mv /mnt/c/Users/Aluno/dmel-gene.fasta ~/myseq/
```

c. Ir para a pasta myseq (cd)

```
cd ~/myseq/
```

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

d. Copiar o arquivo com novo nome: novo.fasta

e. Apagar o arquivo novo.fasta

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

d. Copiar o arquivo com novo nome: novo.fasta

```
cp dmel-gene.fasta novo.fasta
```

e. Apagar o arquivo novo.fasta

UNIX

Comandos mais utilizados

3. Criando e alterando arquivos e diretórios

d. Copiar o arquivo com novo nome: novo.fasta

```
cp dmel-gene.fasta novo.fasta
```

e. Apagar o arquivo novo.fasta

```
rm novo.fasta
```

