

Unidade I - Introdução ao Linux e à Linguagem C

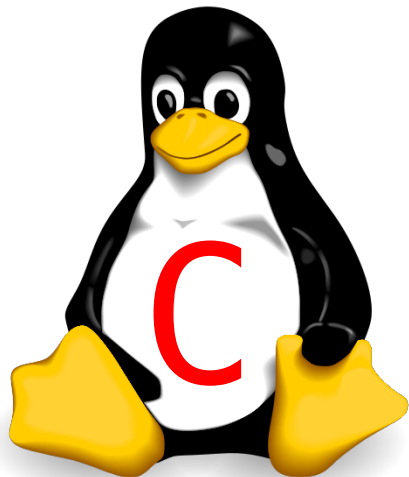
Disciplina Linguagens de Programação I
Bacharelado em Ciência da Computação da Uerj
Professores Guilherme Abelha & Gilson Costa

ANSI C

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

Que assuntos serão abordados nesta unidade?

- Sistema Linux:
 - arquitetura
 - sistema de arquivos
 - uso desktop
 - shell
 - instalação
- Conceitos básicos da linguagem C:
 - Função `main()`
 - Declaração de variáveis
 - Comando `if`
 - Comando `for`
 - `printf()/scanf()`
 - Funções
 - Primeiro programa
 - Compilação Básica
 - Execução do programa
 - Ambiente VPL (Moodle)



Introdução ao Linux

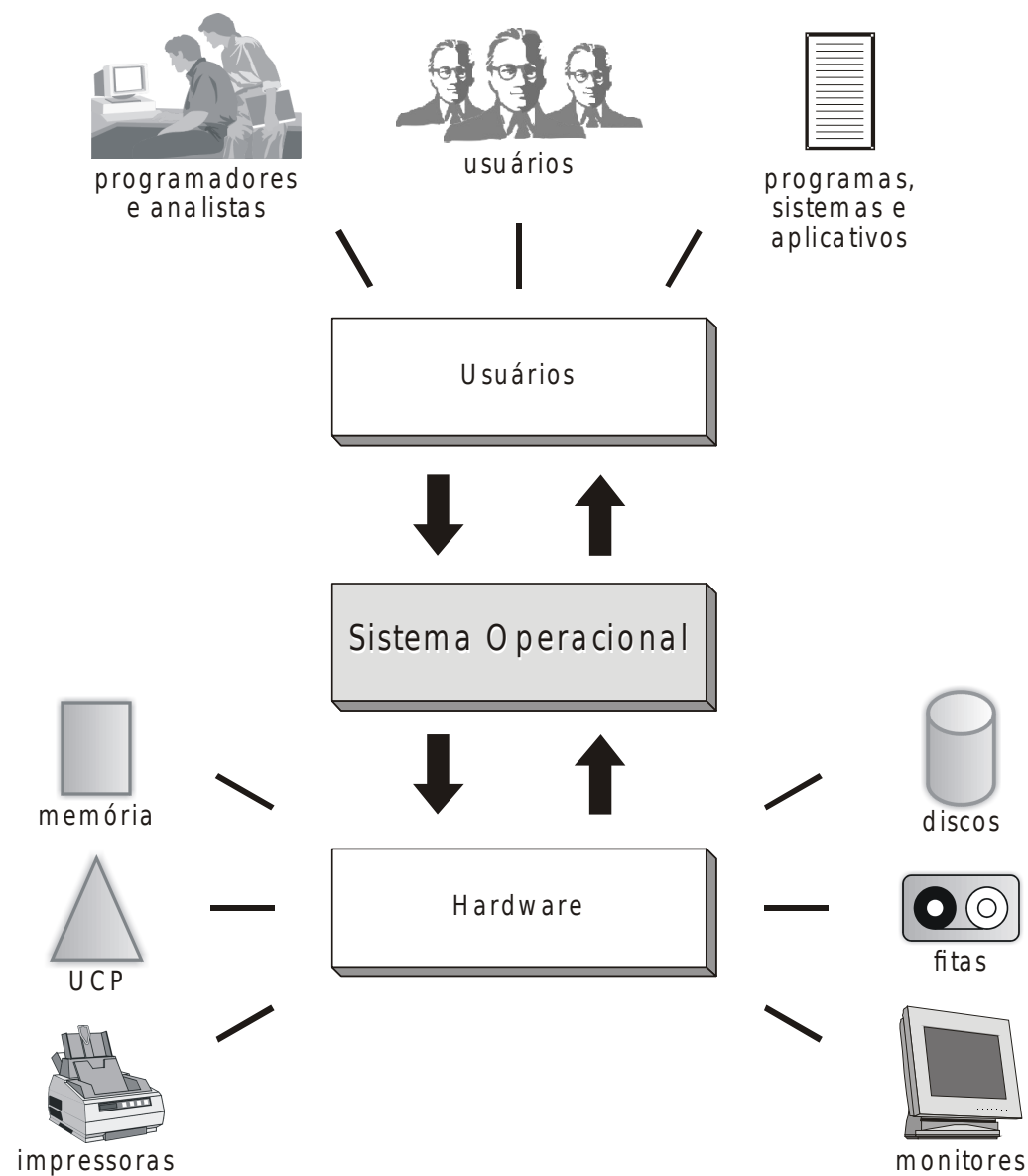
Sugestão de leitura:

Manual Completo do Linux – Guia do Administrador

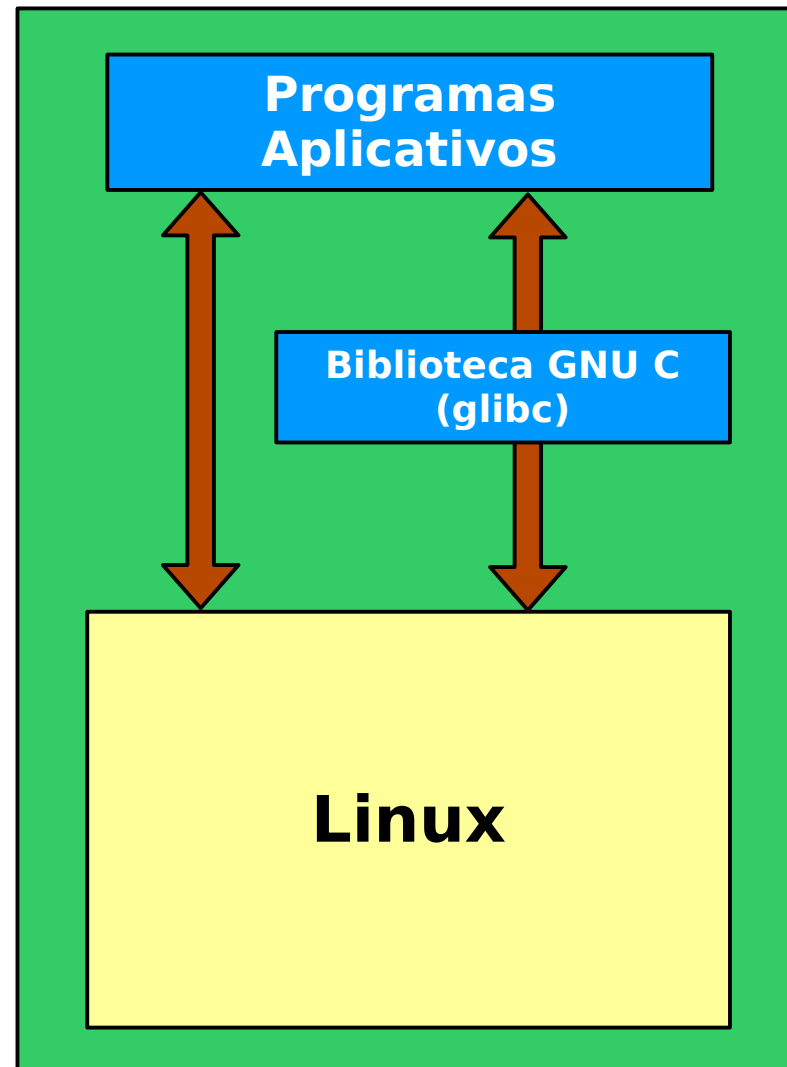
Evi Nemeth/ Garth Snyder/ Trent R. Hein

2ª Edição Pearson / Makron Books

Visão Geral de um Sistema Operacional



Visão Geral da Arquitetura do Sistema



Estrutura de Diretórios

- / diretório raiz do sistema de arquivos
- /bin executáveis de programas básicos
- /etc arquivos de configuração
- /home arquivos pessoais dos usuários
- /lib bibliotecas de uso geral
- /sbin ferramentas administrativas básicas
- /usr arquivos de uso comum

Terminal Linux

Terminal do Linux - Manipulação de Arquivos

`clear` limpa o terminal

`> teste.txt` cria arquivo vazio

`ls` lista o conteúdo de diretório

`ls -la`

`rm <nome>` remove arquivo

`rm <nome> -R`

`mv` move ou renomeia arquivo

`cp` copia arquivo

`chmod <modo> <arq>` muda permissões

Terminal do Linux - Manipulação de Diretórios

`pwd` informa o diretório atual

`cd` muda diretório atual

`cd ~/`

`cd ../`

`mkdir` cria diretório

`rmdir` remove diretório

Terminal do Linux - Manipulação de Programas

`./progName` **executa programa**

`./progName &`

`ps` **lista processos em execução**

`kill` **derruba uma thread**

`sudo` **executa como se fosse o root**

`man <comando>` **exibe documentação**

Terminal do Linux - Gerenciamento de Pacotes

`apt-get` gerencia a instalação de pacotes

`update` atualiza lista pacotes

`upgrade` atualiza pacotes

`dist-upgrade` atualiza distribuição

`install` baixa e instala pacotes

`remove` desinstala pacotes

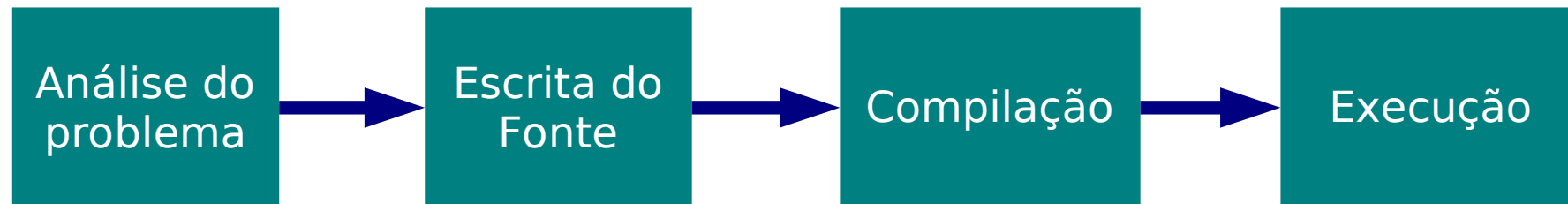
`purge` purga pacotes

Instalação do Linux

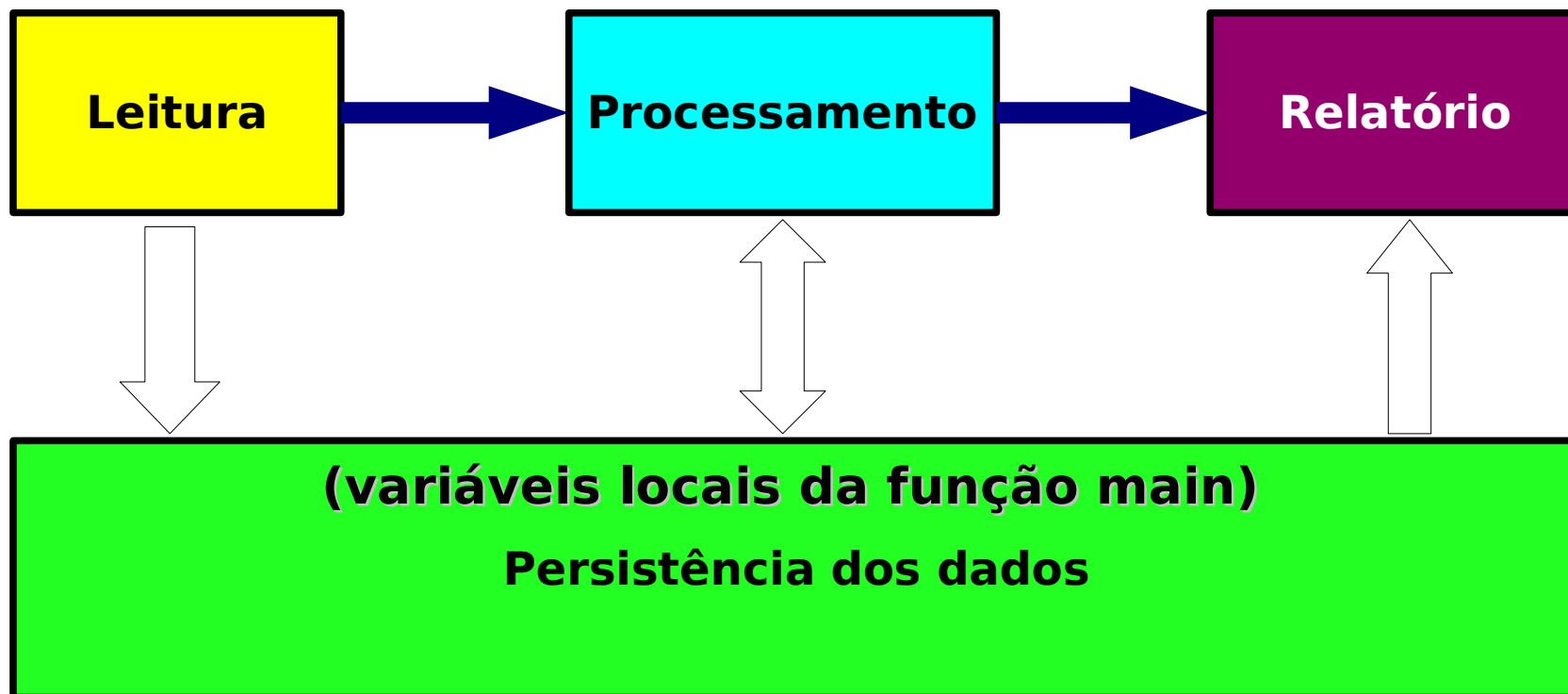
https://linuxmint-installation-guide.readthedocs.io/pt_BR/latest/

Conceitos básicos da linguagem C

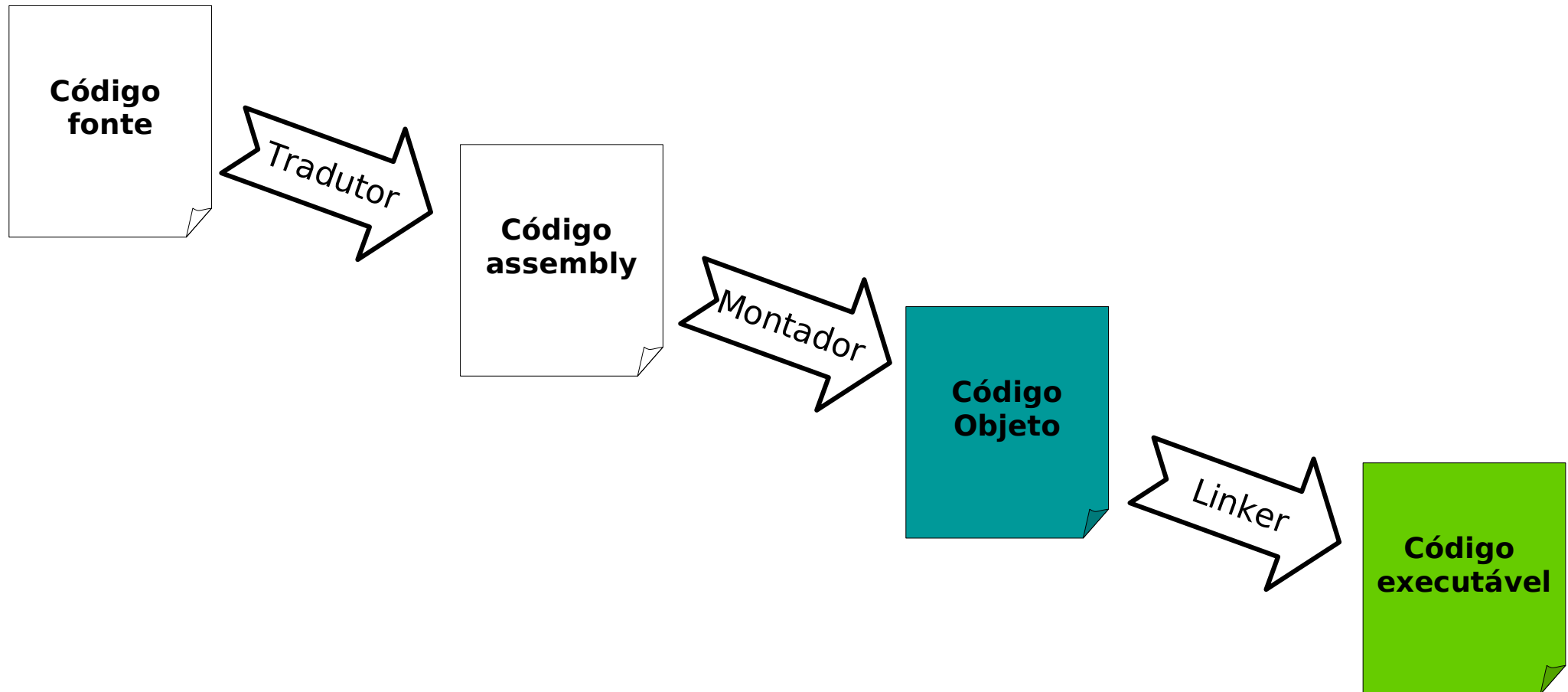
Etapas da elaboração de um programa C



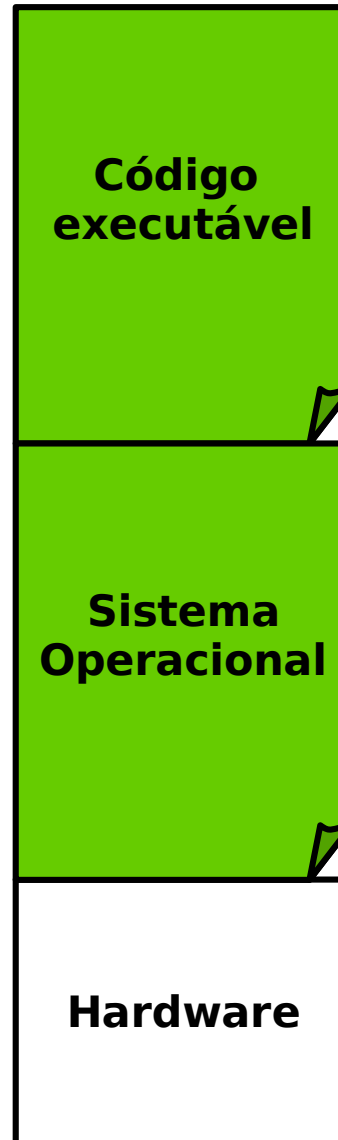
Subdivisão da Computação



Compilador = Tradutor + Montador + Linker



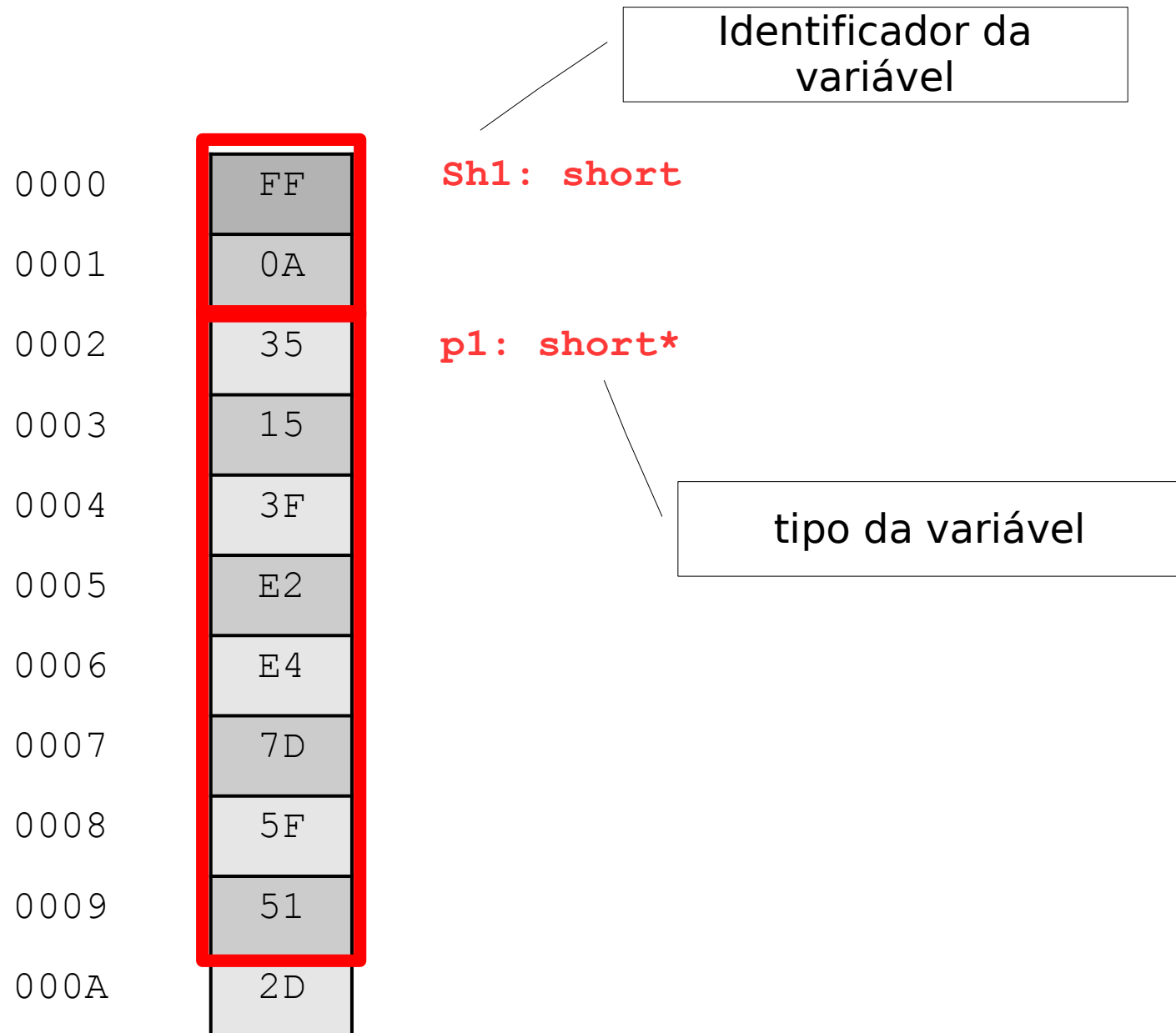
Execução de um programa



Memória Principal

0000	FF	000B	0F	0016	AA
0001	0A	000C	2A	0017	BA
0002	35	000D	E5	0018	55
0003	15	000E	35	0019	43
0004	3F	000F	6F	001A	54
0005	E2	0010	B2	001B	38
0006	E4	0011	E3	001C	B6
0007	7D	0012	73	001D	5A
0008	5F	0013	4F	001E	68
0009	51	0014	23	001F	88
000A	2D	0015	14	0020	34

Variáveis



Primeiro Programa

Primeiro Programa

- Escreva um programa que leia dois números inteiros, calcule a divisão e imprima o resultado na tela do terminal.
- É preciso testar se o segundo número é zero.
 - Padronização do programa principal
 - Declaração e uso de variáveis
 - Operações aritméticas
 - Seleção de fluxo
 - Operação de leitura do teclado
 - Operação de escrita na tela

Primeiro Programa

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

Primeiro Programa

Inclusão de
biblioteca

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

Primeiro Programa

Programa
principal

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```


Primeiro Programa

primeiroProg.c

Declaração de
variáveis

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

Primeiro Programa

primeiroProg.c

Operações de
entrada

Operações de
saída

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

Passagem de parâmetro por valor

valor.c

```
#include <stdio.h>

void incrementa (int num);

int main ()
{
    int num = 9;
    incrementa (num);
    printf ("%d", num);
    return 0;
}

void incrementa (int num)
{
    num = num + 1;
}
```

Passagem de parâmetro por Referência

referencia.c

```
#include <stdio.h>

void incrementa (int *);

int main ()
{
    int num = 9;
    incrementa (&num);
    printf ("%d", num);
    return 0;
}

void incrementa (int *num)
{
    *num = *num + 1;
}
```

printf() e scanf()

```
int printf(char* format, arg1, arg2, ...);
```

```
printf("%d\n", c);
```

```
int scanf(char* format, arg1, arg2, ...);
```

```
scanf("%d", &b);
```

Caractere	Tipo do argumento
%d	int
%c	int
%s	string
%f	float

<http://www.cplusplus.com/reference/cstdio/printf/>

Primeiro Programa

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

expressão

Primeiro Programa

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

Seleção de
fluxo

if - else

```
/* expressao != 0 ? */
```

```
    if (expressao)  
        comando-1;  
    else  
        comando-2;
```

```
/* note que é expressão e  
não condição*/
```


Programando no Ambiente Linux

Compilando e executando o primeiro programa

Editor de texto não formatado
gedit (**g**nome text **edit**or)



The screenshot shows a window of the gedit text editor. The title bar indicates the file is 'primeiroProg.c' located in the home directory (~/.). The window contains a C program that reads two integers, a and b, and calculates the division of a by b if b is not zero. The code is as follows:

```
1 #include <stdio.h>
2 int main ()
3 {
4     int a, b, c;
5     scanf("%d",&a);
6     scanf("%d",&b);
7     if (b!=0)
8     {
9         c = a / b;
10        printf("%d\n",c);
11    }
12    else
13        printf("Divisão por zero");
14    return 0;
15 }
```

The status bar at the bottom shows the file encoding is 'C', the tab width is 'Largura da tabulação: 4', the cursor is at 'Lin 15, Col 2', and the input mode is 'INS'.

Compilando e executando o primeiro programa

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

```
> gcc primeiroProg.c
> ./a.out
```

Compilando e executando o primeiro programa

primeiroProg.c

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (b!=0)
    {
        c = a / b;
        printf ("%d\n", c);
    }
    else
        printf ("Divisão por zero");
    return 0;
}
```

```
> gcc primeiroProg.c -o primeiro
> ./primeiro
```

Debugger

Debugger

- Compilando modo debugger

```
> gcc primeiroProg.c -ggdb
```

- Executando o debugger

```
> gdb a.out
```

- Criando um breakpoint

```
> (gdb) break primeiroProg.c:13
```

- Iniciando a execução

```
> (gdb) run
```

Debugger

- Exibindo o conteúdo de uma variável

```
> (gdb) print a  
> (gdb) printf "%d", a
```

- Continuando

```
> (gdb) cont
```

- Saindo

```
> (gdb) quit
```

Primeiro Problema

Primeiro Problema

- Escreva um programa que imprima uma tabela dos valores de temperatura em Fahrenheit e Celsius.
 - Valor inicial: 0 °F
 - Valor Final: 300 °F
 - Incremento: 20 °F

$$C = \frac{5}{9} \times (F - 32)$$

Fahrenheit	Celsius
0	-17.8
20	-6.7
40	4.4
60	15.6
...	...
300	148.9

Programa Fahrenheit Celsius

```
#include <stdio.h>

// protótipo da função
float fahr2cels (int);

int main()
{
    int fahr;
    for (fahr = 0; fahr <= 300; fahr = fahr + 20)
        printf("%d %f\n", fahr, fahr2cels(fahr));
    return 0;
}

// implementação da função
float fahr2cels (int fahr)
{
    return (5.0 / 9) * (fahr - 32);
}
```

```
$ ./progPag026
0 -17.8
20 -6.7
40 4.4
60 15.6
80 26.7
100 37.8
120 48.9
140 60.0
160 71.1
180 82.2
200 93.3
220 104.4
240 115.6
260 126.7
280 137.8
300 148.9
$
```

Programa Fahrenheit Celsius Ingênuo

```
#include <stdio.h>
int main()
{
    printf("%d %f\n", 0, -17.8);
    printf("%d %f\n", 20, -6.7);
    printf("%d %f\n", 40, 4.4);
    printf("%d %f\n", 60, 15.6);
    printf("%d %f\n", 80, 26.7);
    printf("%d %f\n", 100, 37.8);
    printf("%d %f\n", 120, 48.9);
    printf("%d %f\n", 140, 60.0);
    printf("%d %f\n", 160, 71.1);
    printf("%d %f\n", 180, 82.2);
    printf("%d %f\n", 200, 93.3);
    printf("%d %f\n", 220, 104.4);
    printf("%d %f\n", 240, 115.6);
    printf("%d %f\n", 260, 126.7);
    printf("%d %f\n", 280, 137.8);
    printf("%d %f\n", 300, 148.9);
    return 0;
}
```

```
$ ./progPag026
0 -17.8
20 -6.7
40 4.4
60 15.6
80 26.7
100 37.8
120 48.9
140 60.0
160 71.1
180 82.2
200 93.3
220 104.4
240 115.6
260 126.7
280 137.8
300 148.9
$
```

O comando for

```
for (pré-execução; condição; pós-iteração)
{
    ... ;
}
```

Declaração de uma função

- Protótipo da função

```
Tipo_Retorno nome_func(Tipos_Args);
```

- Implementação da função

```
Tipo_Retorno nome_func(argumentos)
{
    Declarações e instruções;
    return expressão;
}
```

Resultado do Programa Fahrenheit Celsius

```
#include <stdio.h>

// protótipo da função
float fahr2cels (int);

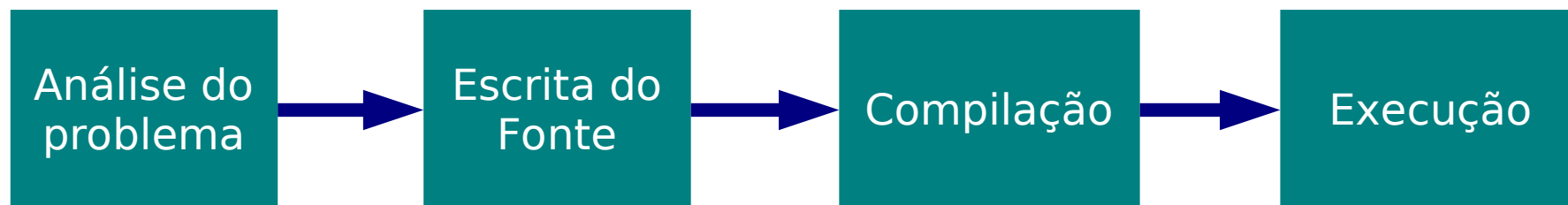
int main()
{
    int fahr;
    for (fahr = 0; fahr <= 300; fahr = fahr + 20)
        printf("%d %f\n", fahr, fahr2cels(fahr));
    return 0;
}

// implementação da função
float fahr2celc (int fahr)
{
    return (5.0 / 9) * (fahr - 32);
}
```

```
$ ./progPag026
0 -17.8
20 -6.7
40 4.4
60 15.6
80 26.7
100 37.8
120 48.9
140 60.0
160 71.1
180 82.2
200 93.3
220 104.4
240 115.6
260 126.7
280 137.8
300 148.9
$
```

Exercício U1.1

- Escreva, utilizando o comando `for`, um programa que imprima a tabela de temperatura de Celsius para Fahrenheit
 - Valor inicial: -10 °C
 - Valor Final: 50 °C
 - Incremento: 10 °C
- Submeta sua solução no moodle



FIM