

MC102 - Algoritmos e Programação de Computadores - (Lab. 1)

IC – UNICAMP

Professor: João Paulo Papa

Monitor: Javier A. Montoya Zegarra-Farach

1º Semestre de 2006

1 Objetivos

1. Familiarizar se com o entorno de programação da linguagem C no sistema operacional linux.
2. Editar, compilar e executar programas na linguagem C.
3. Apresentar as funções de entrada e saída funções printf() e scanf() da linguagem C.
4. Manipulação de variáveis.

2 Getting Started

O objetivo desta seção é a de apresentar passo a passo as atividades a serem desenvolvidas para compilar um programa na linguagem C no sistema operacional linux. Este passos podem ser resumidos da seguinte maneira:

1. Abrir um terminal (Start > System Tools > Terminal) (Veja figura 1).
2. Criar uma pasta na qual os programas serão colocados (mkdir mc102, Veja figuras 2 - 8).
3. Abrir um editor para digitar o programa (Veja figuras 9 - 11).
4. Uma vez digitado o programa, salve-o com a extensão ".c" (Veja figuras 12, 13).
5. Abra um terminal novo na mesma pasta onde o seu arquivo fica.
6. Compile o seu programa (gcc nome.c -o nome) (Veja figuras 14 - 15).
7. Execute o seu programa (./nome) (Veja figura 16).

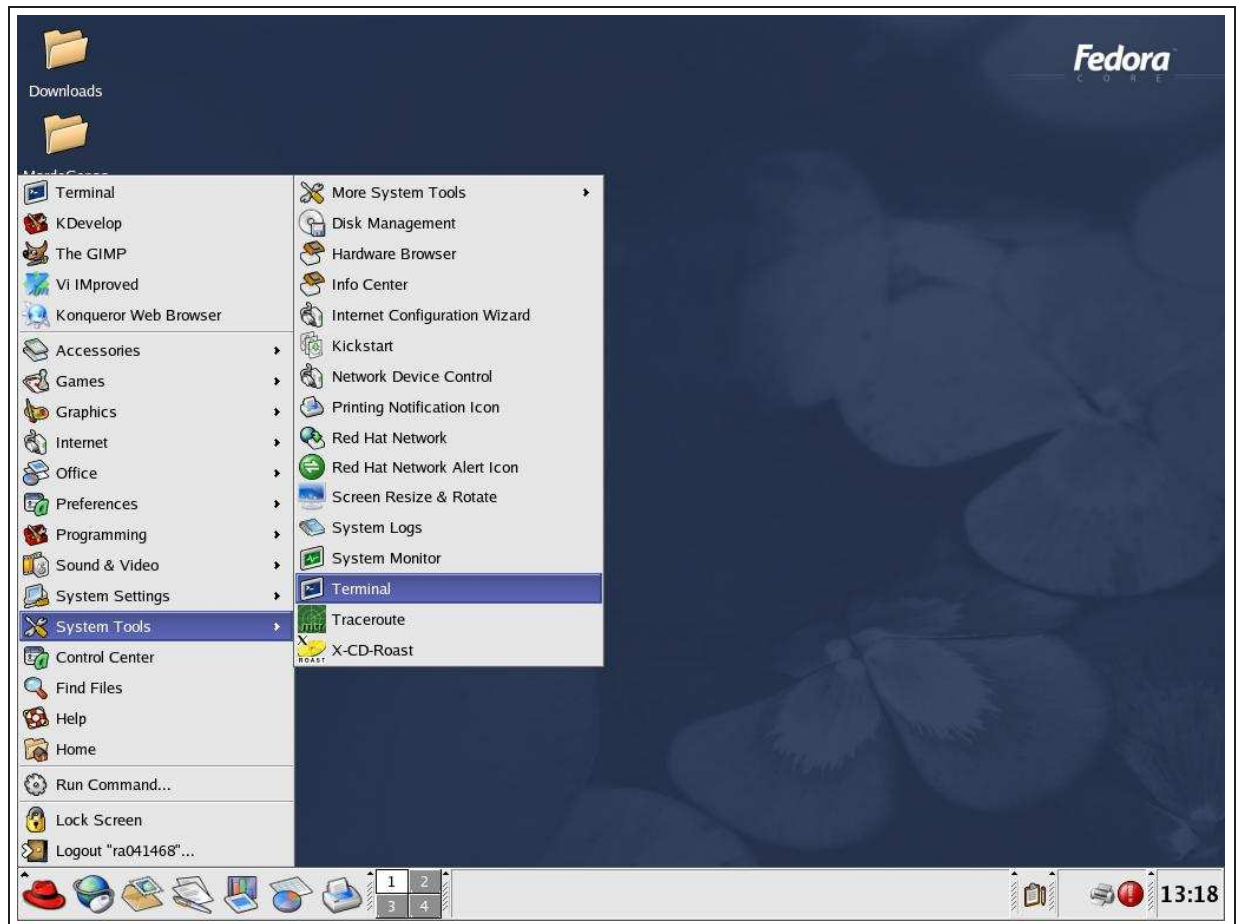


Figura 1: Abrindo um terminal.

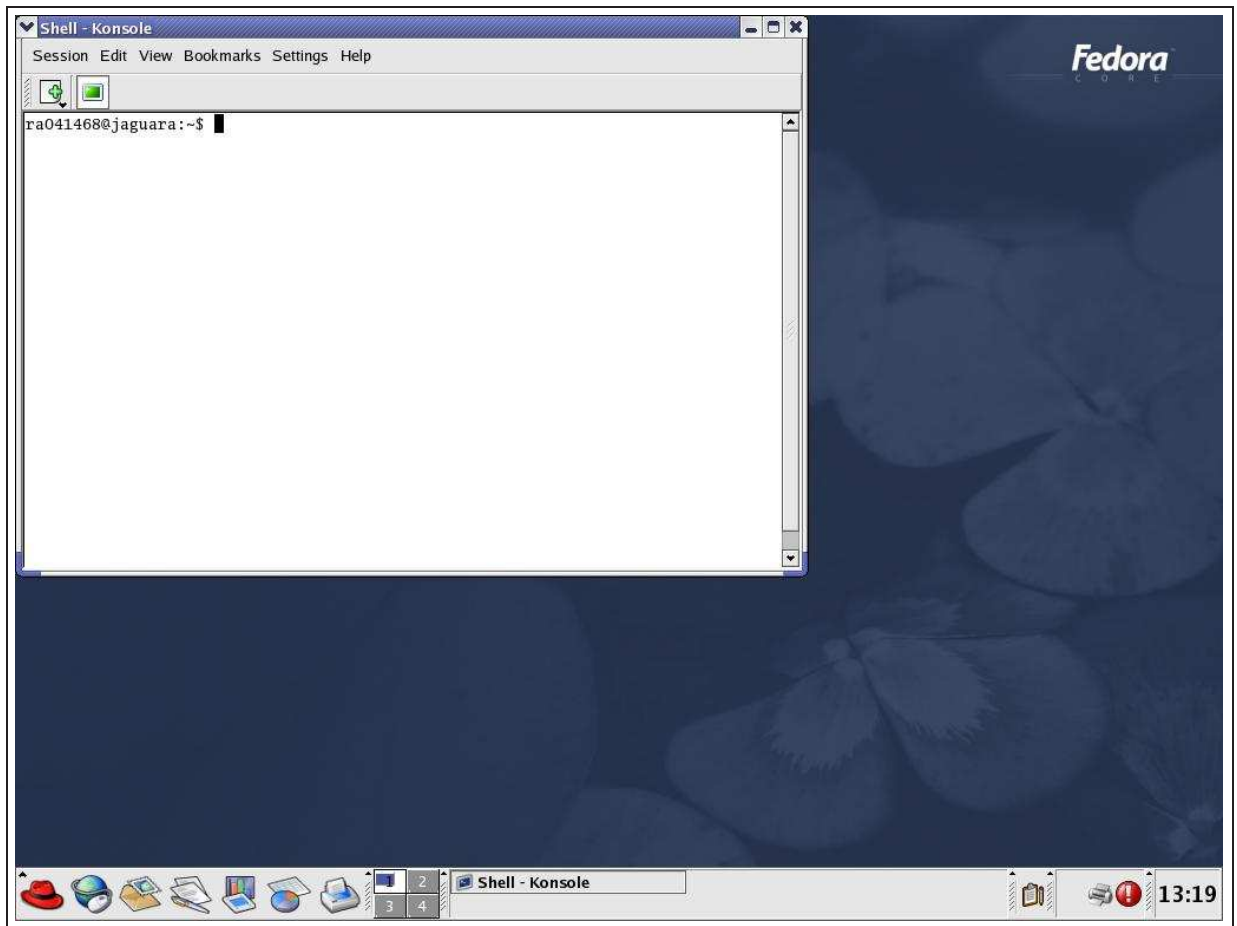


Figura 2: Terminal aberto.

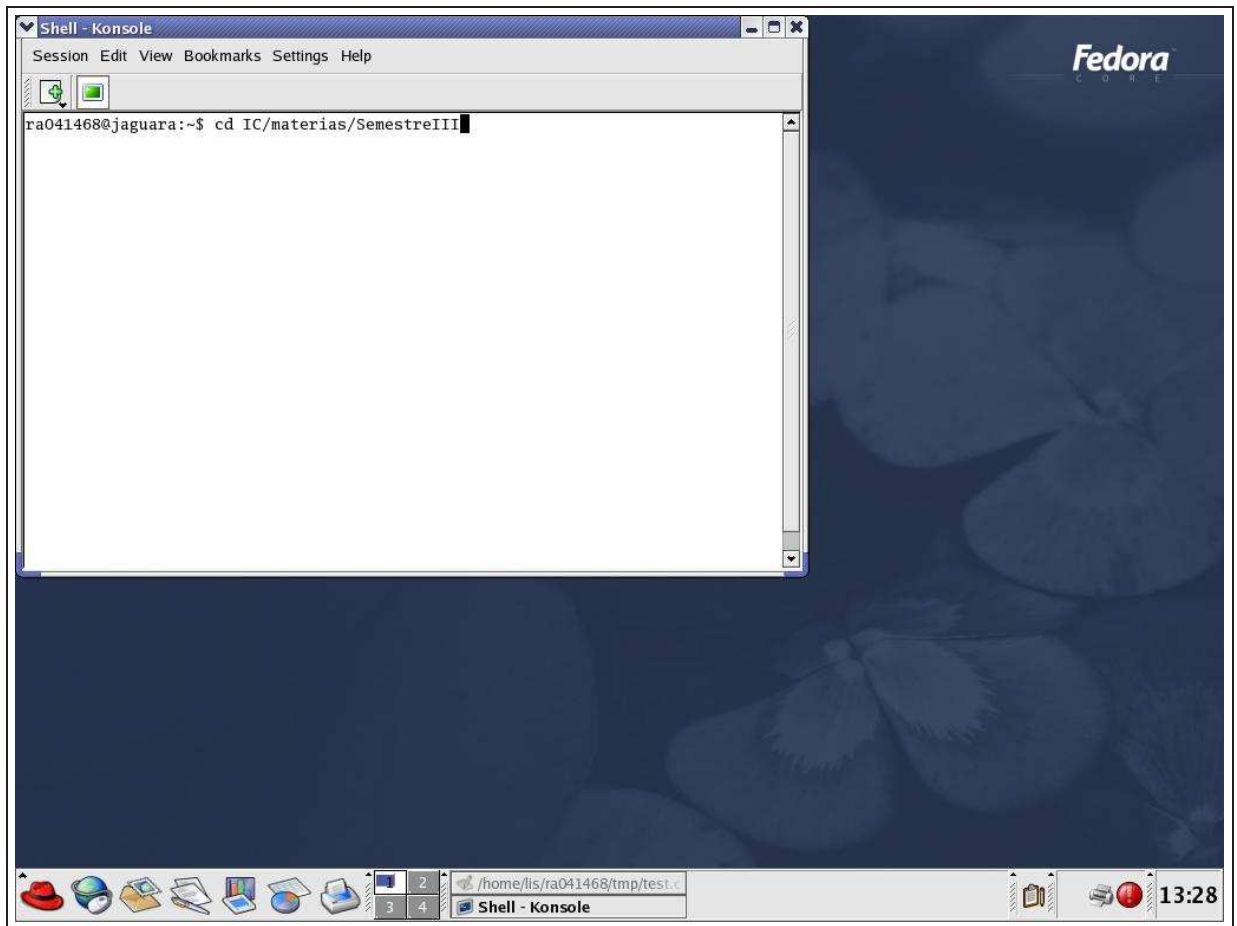


Figura 3: Indo para a pasta SemestreIII.

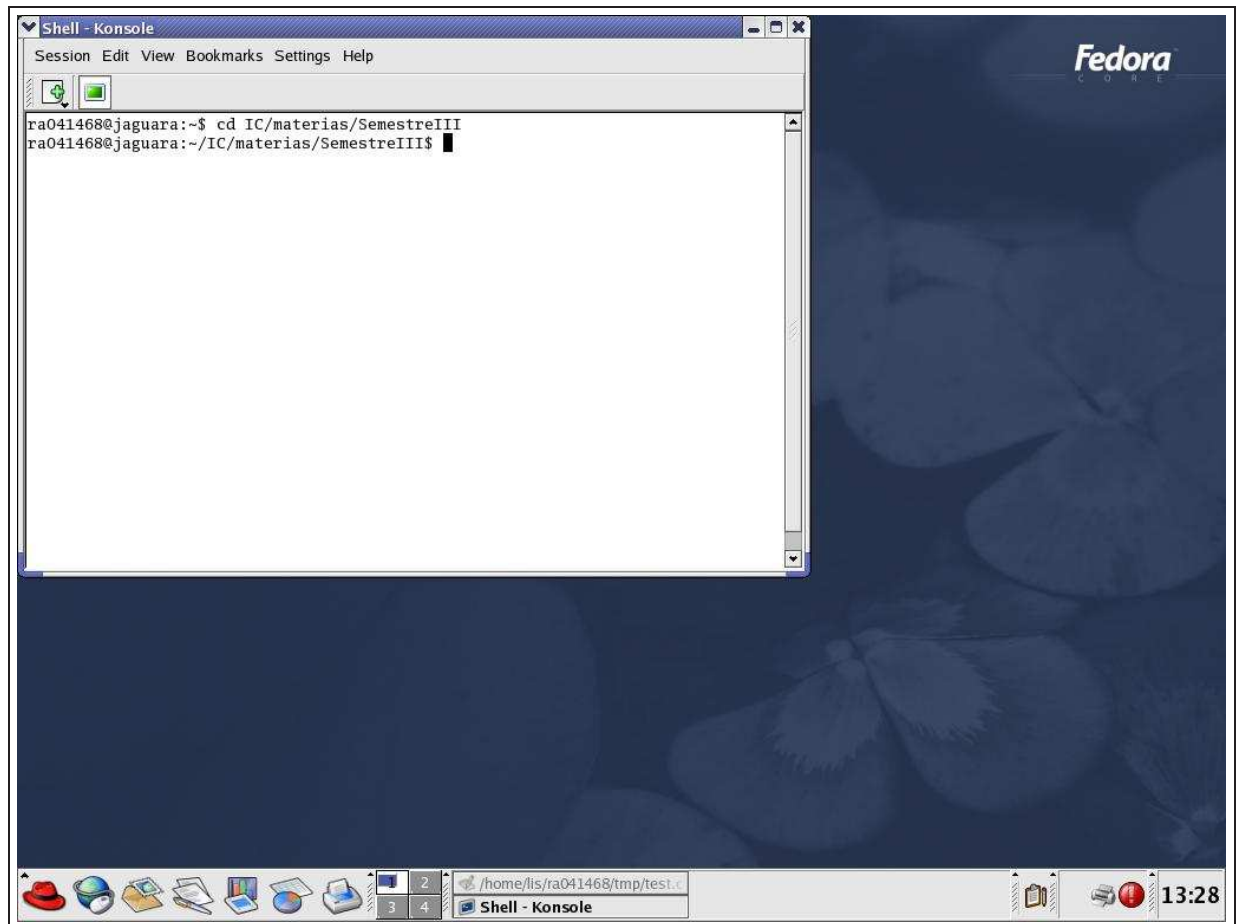


Figura 4: Pasta SemestreIII.

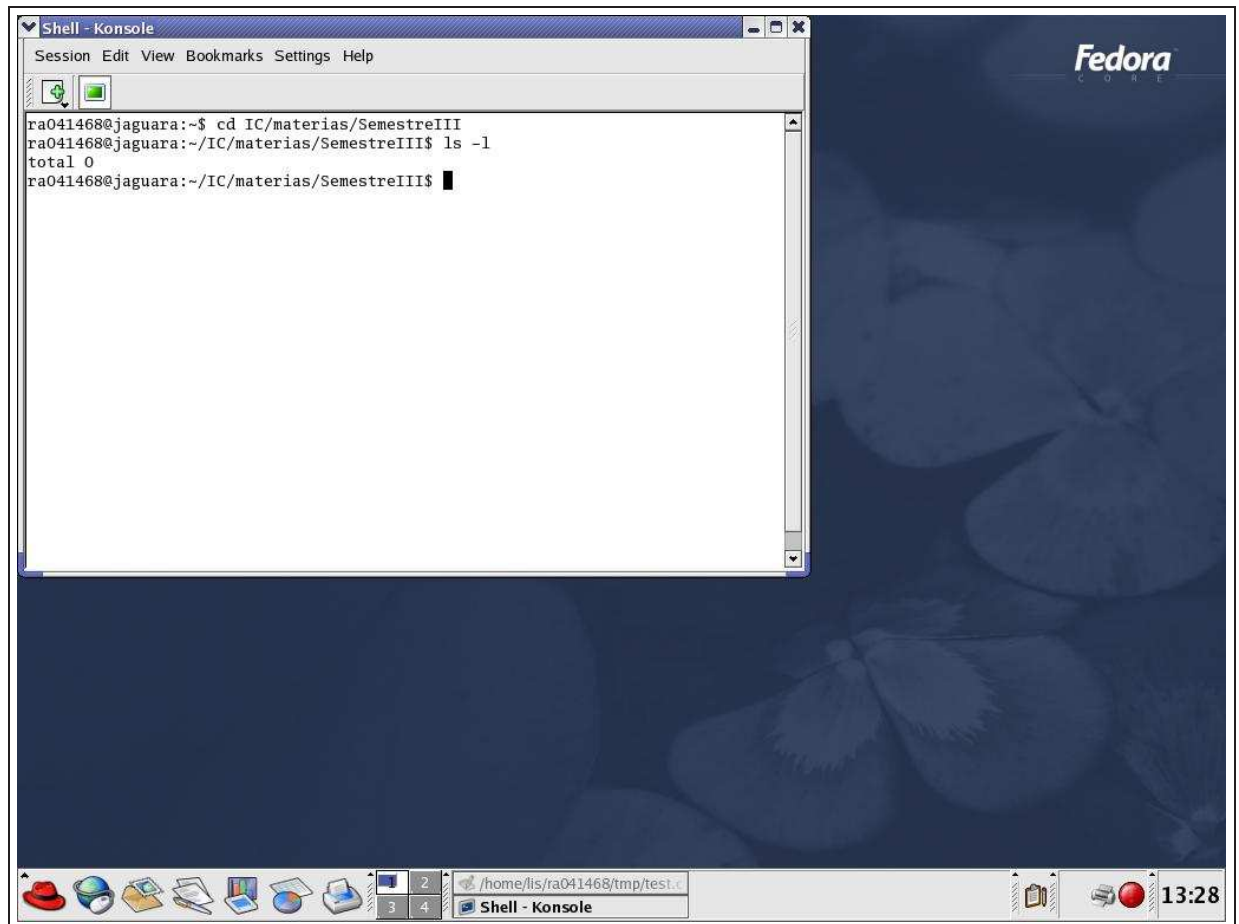


Figura 5: Mostrando o conteúdo da pasta SemestreIII.

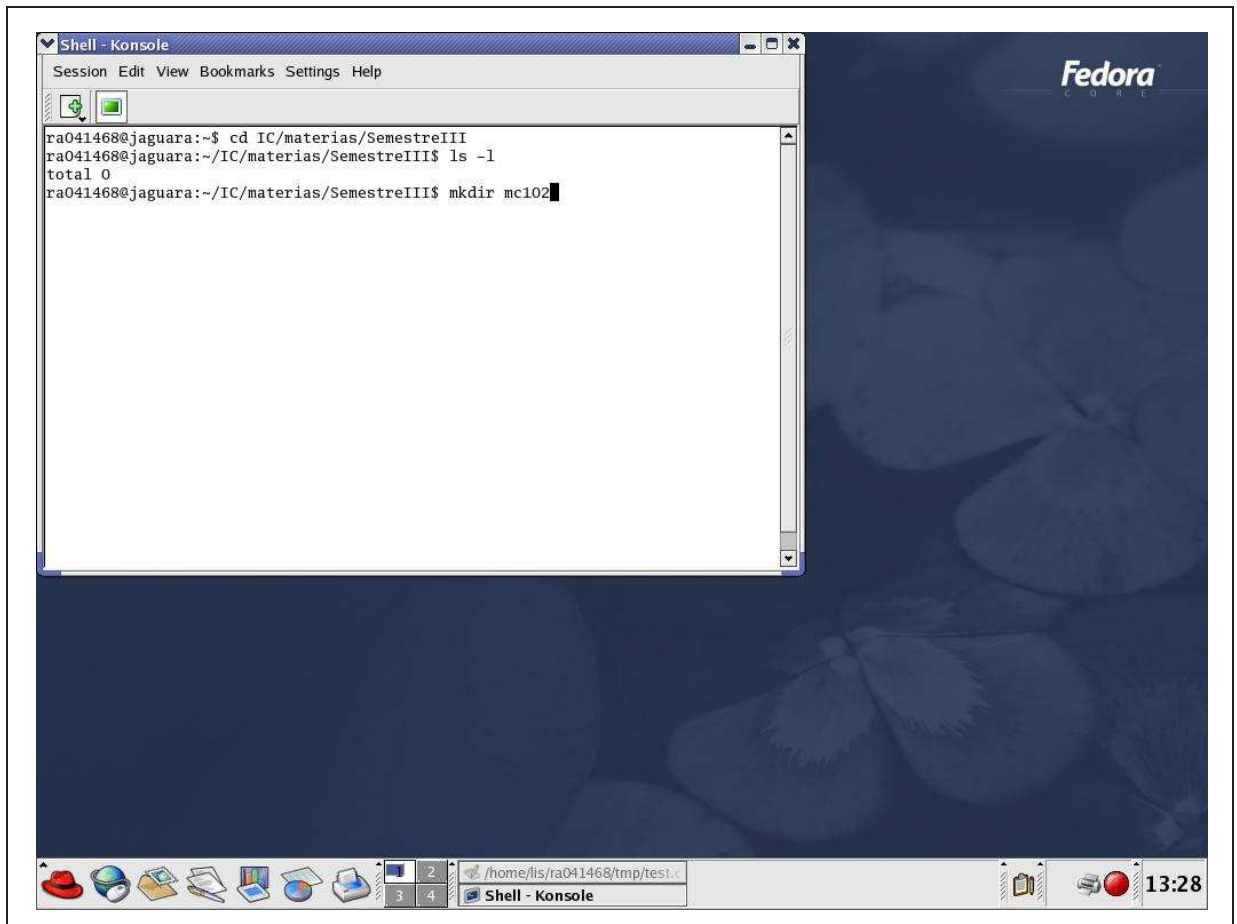


Figura 6: Criando a pasta mc102.

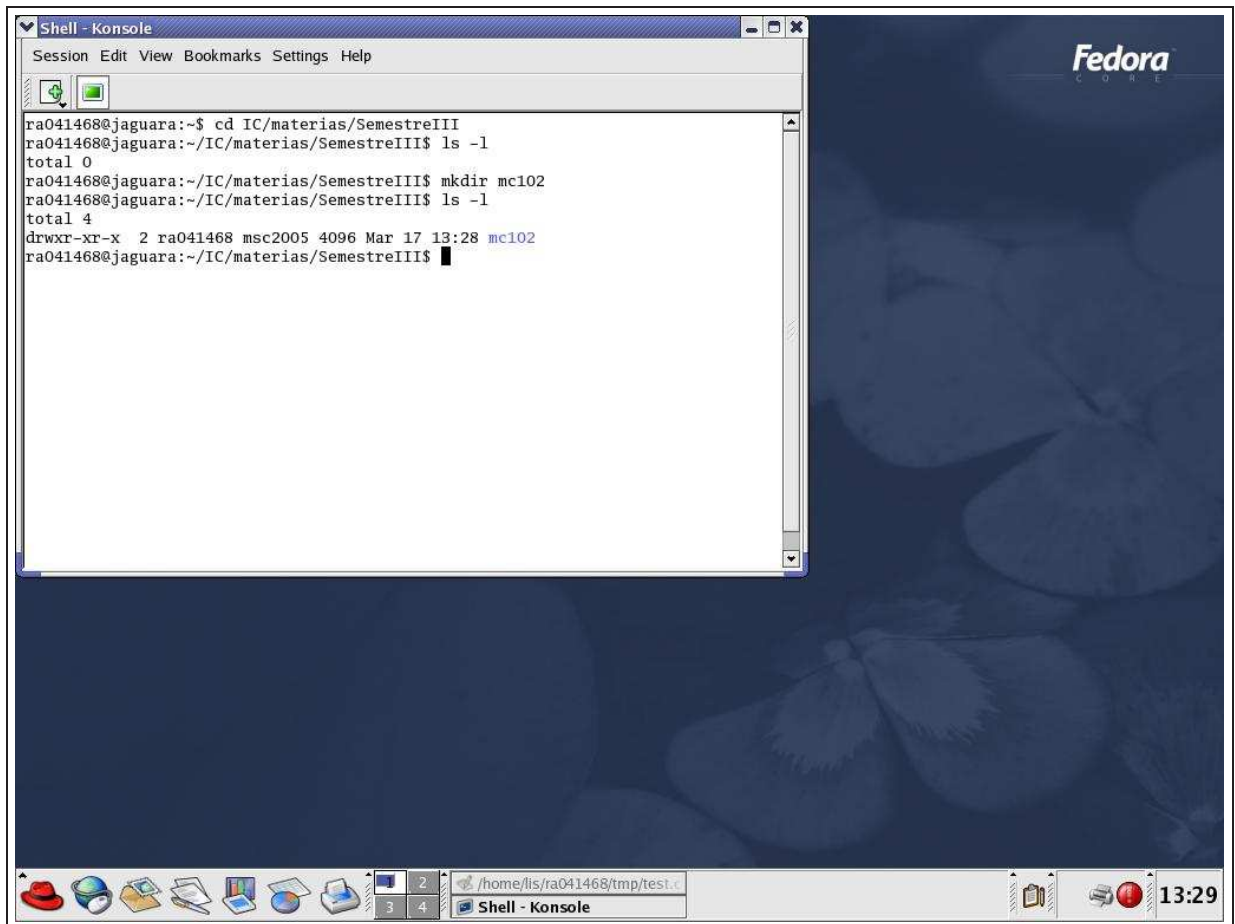


Figura 7: Mostrando o conteúdo da pasta mc102.

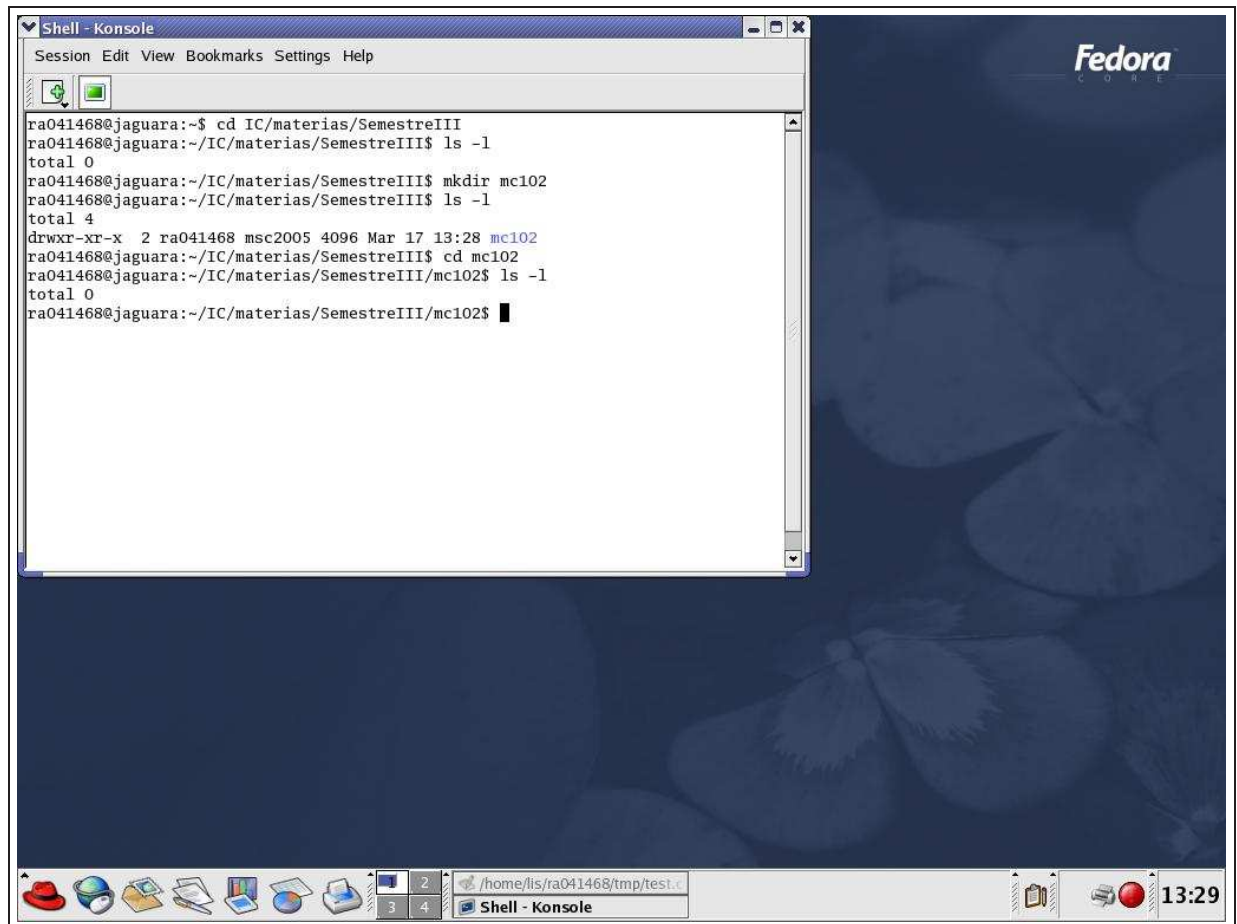


Figura 8: Mostrando o conteúdo da pasta mc102.

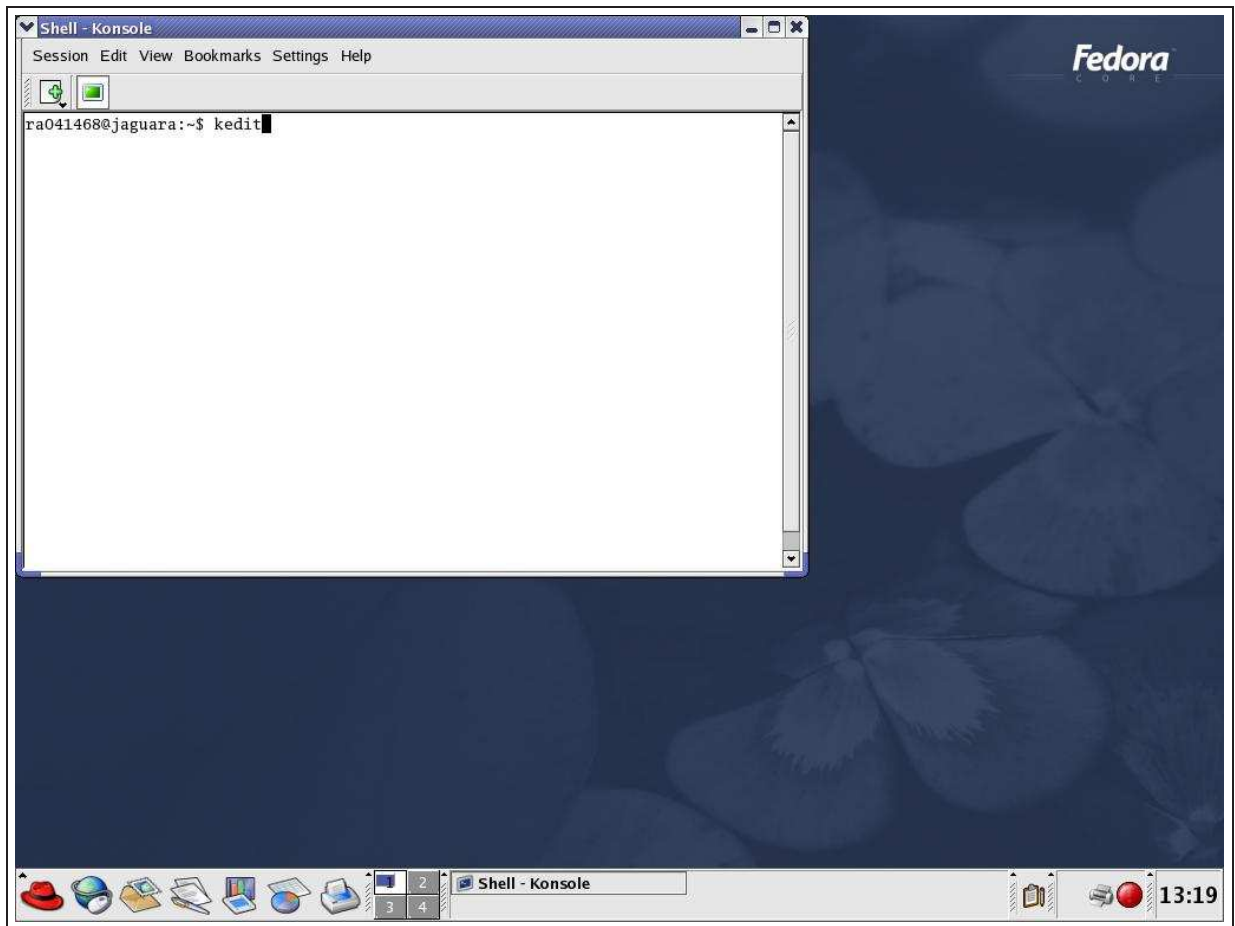


Figura 9: Abrindo o KEdit desde o terminal.

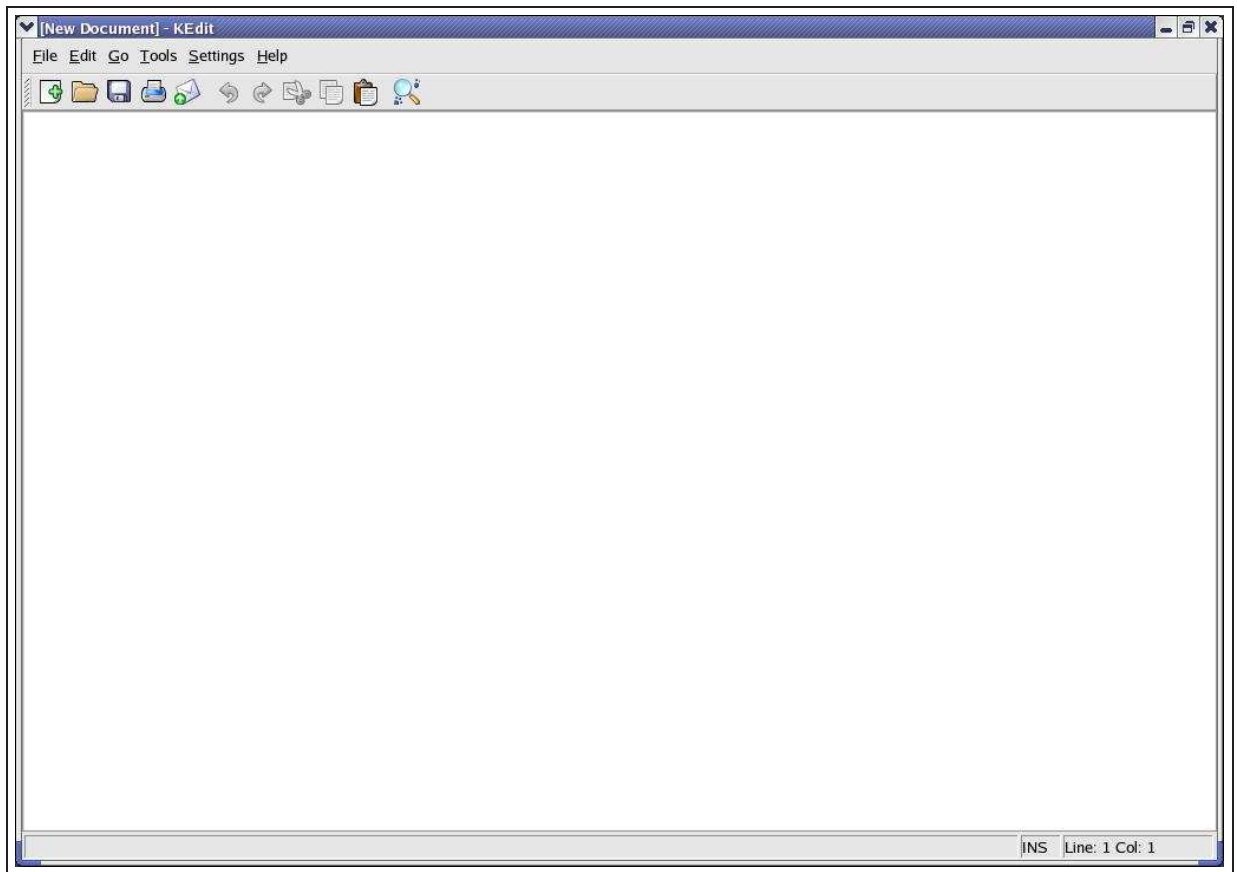
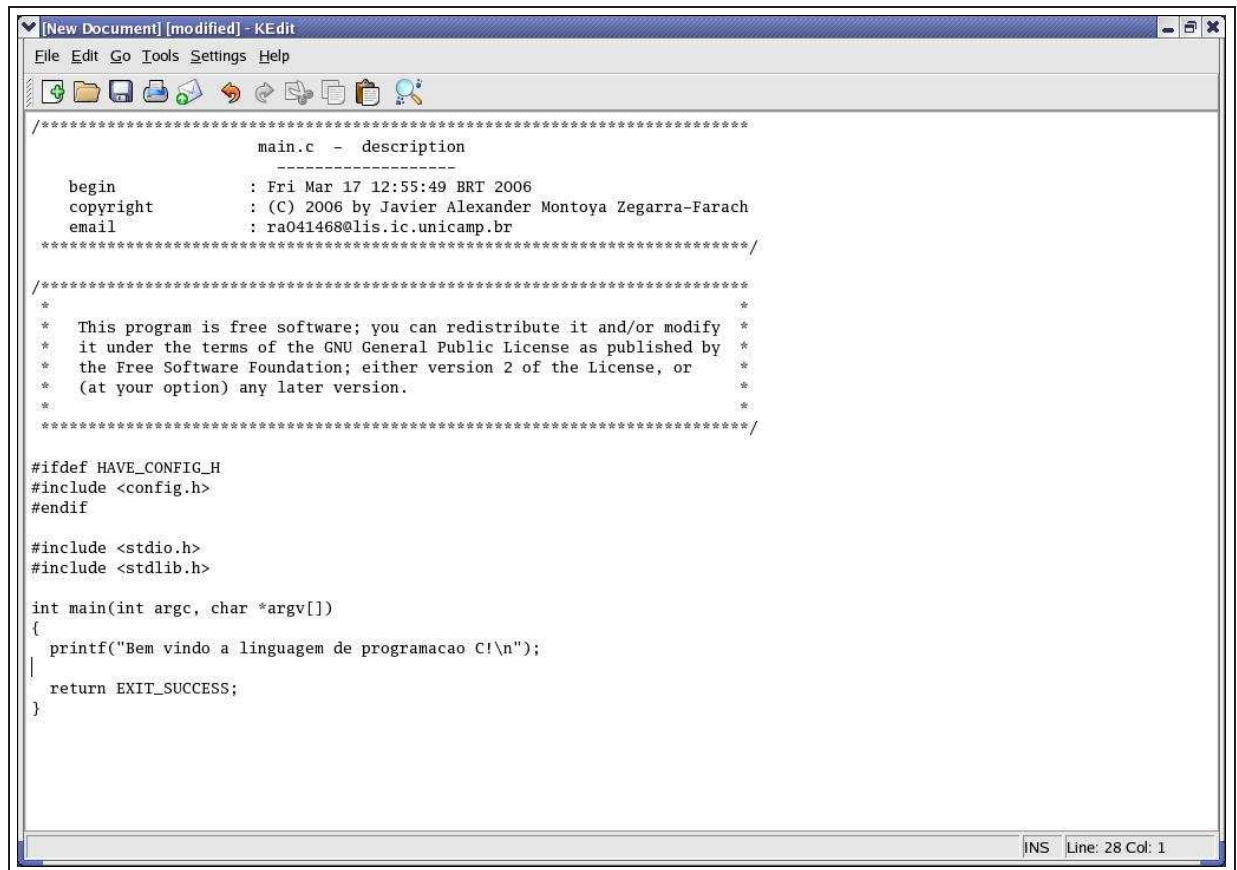


Figura 10: KEdit foi aberto.



The image shows a screenshot of the KEdit text editor window. The title bar reads "[New Document] [modified] - KEdit". The menu bar includes File, Edit, Go, Tools, Settings, and Help. The toolbar contains icons for file operations and editing. The main text area contains a C program template for "main.c". The code includes a header section with a description, begin/end markers, and metadata (date, copyright, email). It also includes a license notice for the GNU General Public License. The main function is defined with includes for <config.h>, <stdio.h>, and <stdlib.h>, and a simple printf statement and return statement.

```
main.c - description
-----
begin      : Fri Mar 17 12:55:49 BRT 2006
copyright  : (C) 2006 by Javier Alexander Montoya Zegarra-Farach
email      : ra041468@lis.ic.unicamp.br
*****/

/*****
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 *****/

#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Bem vindo a linguagem de programacao C!\n");
    return EXIT_SUCCESS;
}
```

INS Line: 28 Col: 1

Figura 11: Criando o nosso primer programa.

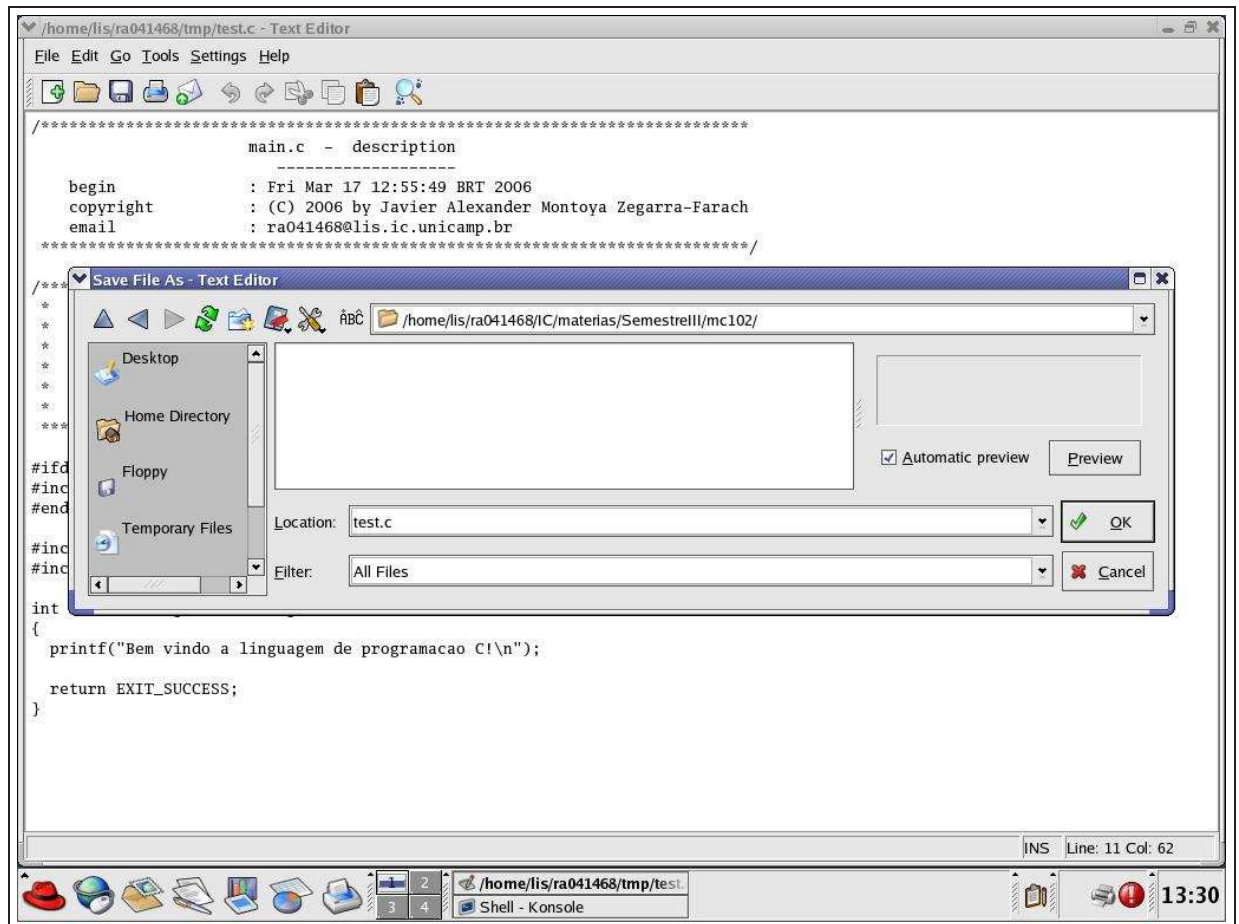
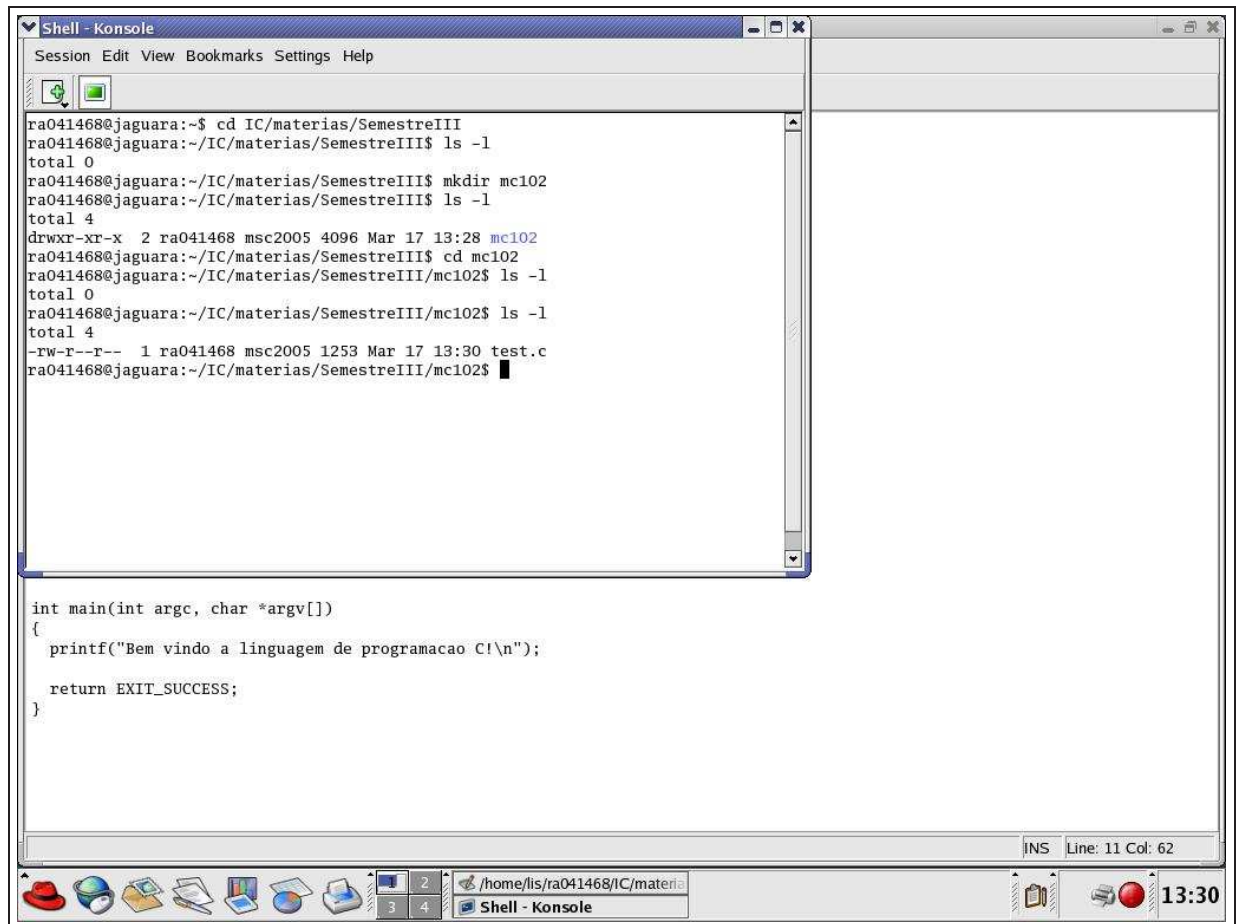


Figura 12: Salvando o nosso programa.



```
ra041468@jaguara:~$ cd IC/materias/SemestreIII
ra041468@jaguara:~/IC/materias/SemestreIII$ ls -l
total 0
ra041468@jaguara:~/IC/materias/SemestreIII$ mkdir mc102
ra041468@jaguara:~/IC/materias/SemestreIII$ ls -l
total 4
drwxr-xr-x  2 ra041468 msc2005 4096 Mar 17 13:28 mc102
ra041468@jaguara:~/IC/materias/SemestreIII$ cd mc102
ra041468@jaguara:~/IC/materias/SemestreIII/mc102$ ls -l
total 0
ra041468@jaguara:~/IC/materias/SemestreIII/mc102$ ls -l
total 4
-rw-r--r--  1 ra041468 msc2005 1253 Mar 17 13:30 test.c
ra041468@jaguara:~/IC/materias/SemestreIII/mc102$
```

```
int main(int argc, char *argv[])
{
    printf("Bem vindo a linguagem de programacao C!\n");
    return EXIT_SUCCESS;
}
```

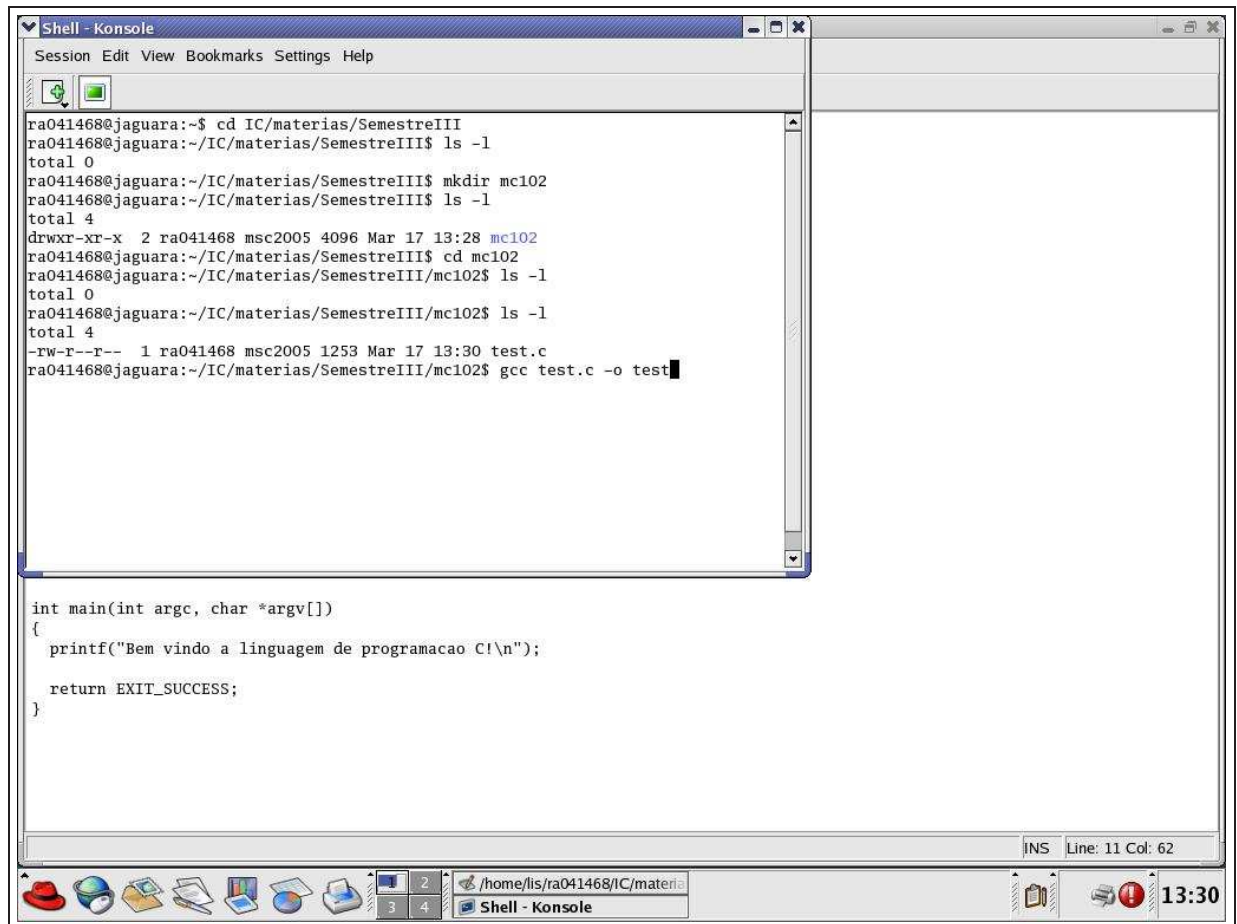
INS Line: 11 Col: 62

/home/lis/ra041468/IC/materias

Shell - Konsole

13:30

Figura 13: Mostrando o conteúdo da pasta mc102 para conferir que o nosso programa foi armazenado.



```
ra041468@jaguara:~$ cd IC/materias/SemestreIII
ra041468@jaguara:~/IC/materias/SemestreIII$ ls -l
total 0
ra041468@jaguara:~/IC/materias/SemestreIII$ mkdir mc102
ra041468@jaguara:~/IC/materias/SemestreIII$ ls -l
total 4
drwxr-xr-x  2 ra041468 msc2005 4096 Mar 17 13:28 mc102
ra041468@jaguara:~/IC/materias/SemestreIII$ cd mc102
ra041468@jaguara:~/IC/materias/SemestreIII/mc102$ ls -l
total 0
ra041468@jaguara:~/IC/materias/SemestreIII/mc102$ ls -l
total 4
-rw-r--r--  1 ra041468 msc2005 1253 Mar 17 13:30 test.c
ra041468@jaguara:~/IC/materias/SemestreIII/mc102$ gcc test.c -o test
```

```
int main(int argc, char *argv[])
{
    printf("Bem vindo a linguagem de programacao C!\n");
    return EXIT_SUCCESS;
}
```

INS Line: 11 Col: 62

/home/lis/ra041468/IC/materias

Shell - Konsole

13:30

Figura 14: Compilando o nosso programa.

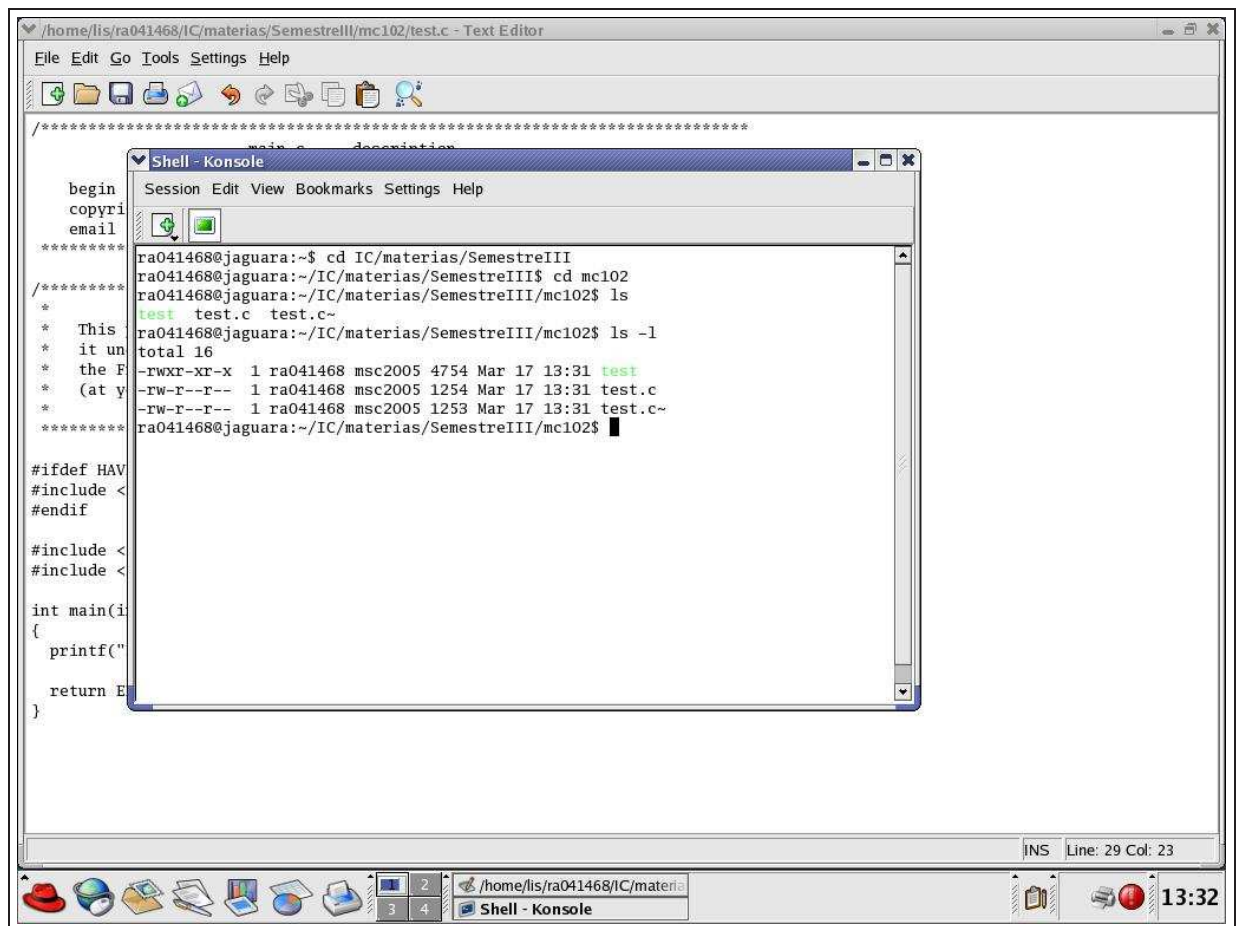


Figura 15: Mostrando o conteúdo da pasta mc1027 para conferir a criação do executável.

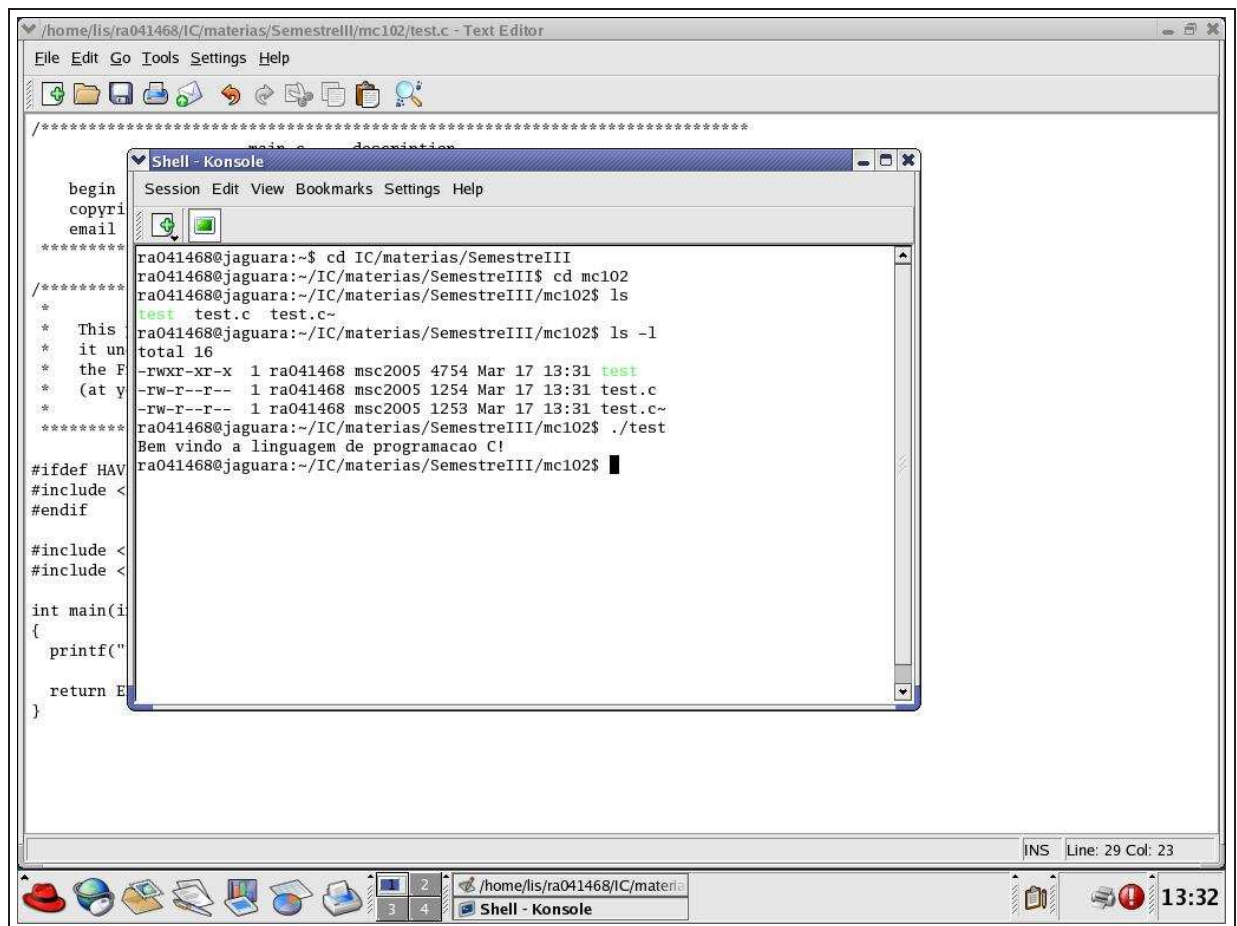


Figura 16: Executando o nosso programa.

3 Exemplos

Nesta seção serão apresentados alguns exemplos de programas.

3.1 Exemplo 1

```
#include <stdio.h>
#include <stdlib.h>

int main ( int argc, char *argv[] )
{
    printf("Oi Mundo Louco!");
    return EXIT_SUCCESS;
}
```

Figura 17: Algoritmo mundo louco

```
$ gcc hello.c -o hello
$ hello
$ Oi Mundo Louco!
```

Figura 18: Executando o programa mundo louco

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Oi mundo louco!\n");

    return EXIT_SUCCESS;
}
```

Figura 19: Algoritmo mundo louco ver. 1.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Oi mundo");
    printf(" louco!\n");

    return EXIT_SUCCESS;
}
```

Figura 20: Algoritmo mundo louco ver. 2.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Oi");
    printf(" mundo");
    printf(" louco!\n");

    return EXIT_SUCCESS;
}
```

Figura 21: Algoritmo mundo louco ver. 3.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Oi\n");
    printf(" mundo\n");
    printf(" louco!\n");

    return EXIT_SUCCESS;
}
```

Figura 22: Algoritmo mundo louco ver. 4.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Oi\n");
    printf("\tmundo\n");
    printf("\t\tlouco!\n");

    return EXIT_SUCCESS;
}
```

Figura 23: Algoritmo mundo louco ver. 5.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("\t\t\tOi\n");
    printf("\t\t\tmundo\n");
    printf("Louco!\n");

    return EXIT_SUCCESS;
}
```

Figura 24: Algoritmo mundo louco ver. 6.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Louco,\n");
    printf("\t\t\tOi\n");
    printf("\t\t\t\tmundo!\n");
    return EXIT_SUCCESS;
}
```

Figura 25: Algoritmo mundo louco ver. 7.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Mundo,");
    printf(" oh");
    printf(" Louco!\n");
    return EXIT_SUCCESS;
}
```

Figura 26: Algoritmo mundo louco ver. 8.

3.2 Exemplo 2

```
#include <stdio.h>
#include <stdlib.h>

int main ( int argc, char *argv[] )
{
    printf("8+7*6");
    return EXIT_SUCCESS;
}
```

Figura 27: Algoritmo 2

```
$ gcc num.c -o num
$ num
$ 8+7*6
```

Figura 28: Executando o algoritmo 2

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("8+7*6\n");
    return EXIT_SUCCESS;
}
```

Figura 29: Algoritmo num ver. 1.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("%d\n",8+7*6);
    return EXIT_SUCCESS;
}
```

Figura 30: Algoritmo num ver. 2.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("%d\n",(8+7)*6);
    return EXIT_SUCCESS;
}
```

Figura 31: Algoritmo num ver. 3.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("%c\n",(8+7)*6);
    return EXIT_SUCCESS;
}
```

Figura 32: Algoritmo num ver. 4.

4 Erros

São dois os tipos de erros que ocorrem no desenvolvimento de um programa: erros de compilação e de execução.

4.1 Erros de Compilação

Acontecem quando o computador não é capaz de executar o código por causa de um erro de sintaxe no programa.

```
#include <stdio.h>
#include <stdlib.h>

int main ( int argc, char *argv[] )
{
    printf("Oi Mundo Louco!");
    return EXIT_SUCCESS;
}
```

Figura 33: Algoritmo mundo louco

```
$ gcc hello.c -o hello
$ hello.c: In function 'main':
$ hello.c:5: error: syntax error at end of input
```

Figura 34: Executando o programa mundo louco

4.2 Erros de Execução

Acontecem quando o comportamento do programa diverge do esperado.

```
#include <stdio.h>
#include <stdlib.h>

int main ( int argc, char *argv[] )
{
    printf("Oi Mundo Louco #%s#%s!");
    return EXIT_SUCCESS;
}
```

Figura 35: Algoritmo mundo louco

```
$ gcc hello.c -o hello
$ hello
$ Oi Mundo Louco #%s#%s!
```

Figura 36: Executando o programa mundo louco