

实验二：C 编程环境

Hollow Man

实验目的

1. 熟悉 Linux 下 C 程序设计的环境；
2. 对系统调用有初步了解。

实验时间

3 学时

实验内容

1. Linux 下 C 语言程序的开发过程

a、在用户主目录下用 vi 编辑 C 语言源程序（源程序已附后），如：**\$vi hello.c**。



```
hollowman@localhost:~  
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)  
[hollowman@localhost ~]$ vi hello.c  
[hollowman@localhost ~]$
```

b、用 gcc 编译 C 语言源程序：**\$gcc ./hello.c -o example**

这里 gcc 是 Linux 下的 C 语言程序编译器（GNU C Compiler），**./hello.c** 表示待编译的源文件是当前工作目录下的 hello.c，**-o example** 表示编译后产生的目标代码文件名为 example。



```
hollowman@localhost:~  
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)  
[hollowman@localhost ~]$ vi hello.c  
[hollowman@localhost ~]$ gcc ./hello.c -o example  
[hollowman@localhost ~]$ ls  
公共 模板 视频 图片 文档 下载 音乐 桌面 example hello.c  
[hollowman@localhost ~]$
```

c、若编译不正确，则进入 vi 修改源程序，否则，运行目标代码：**\$/example**。

注意：

- a、这只是 gcc 最基本的用法，其他常用选项有：**-c** , **-S** , **-O** , **-O2** , **-g** 等。
- b、调试程序可以用 gdb（GNU debugger）。

2. 编辑、调试下面 c 语言程序，说明该程序的功能。

附：hello.c源程序

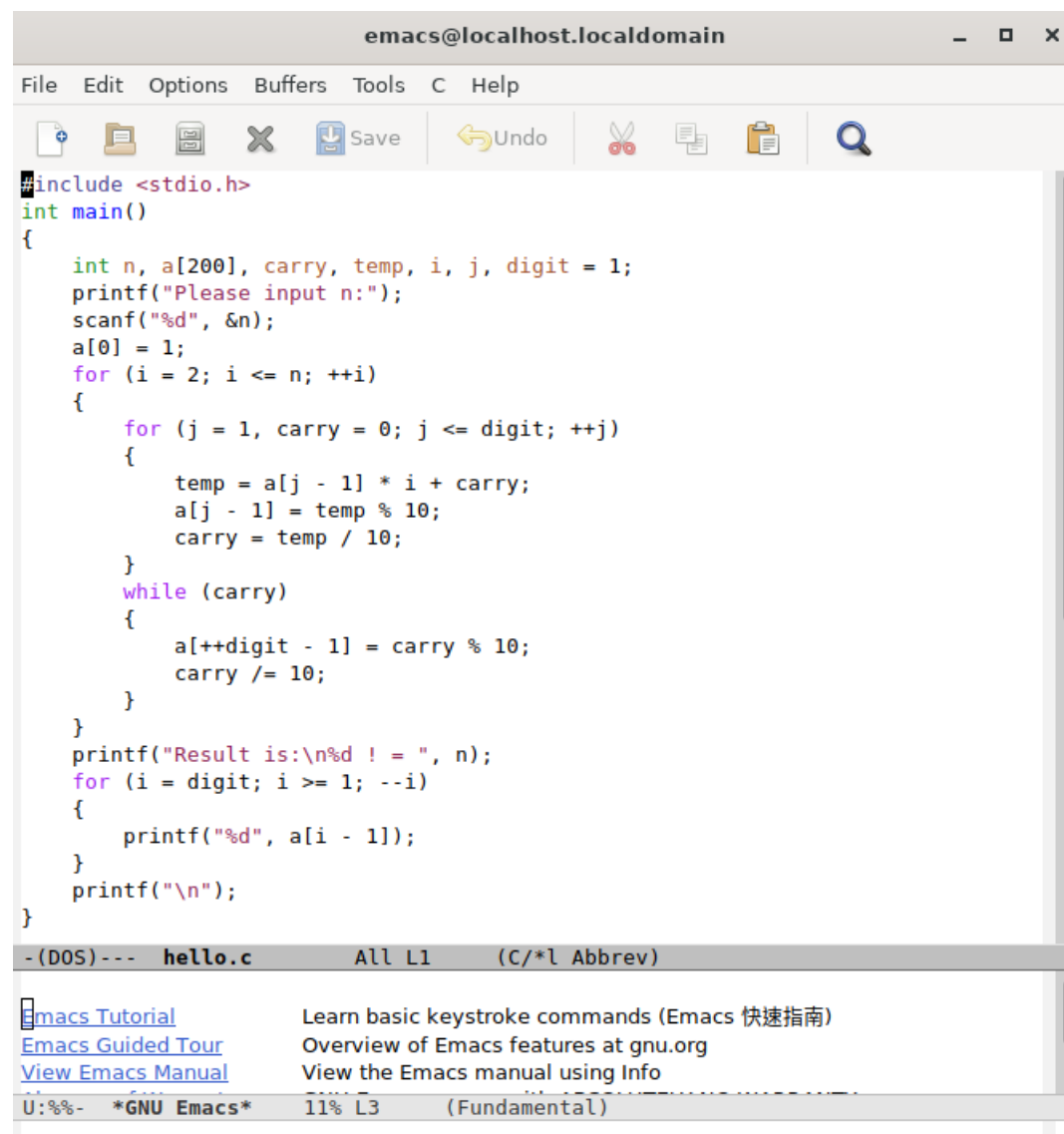
功能：本程序实现了计算输入数字阶乘的功能

代码与注释如下：

```
#include <stdio.h>

main()
{
    int n,a[200],carry,temp,i,j,digit = 1;
    printf("Please input n:");
    scanf("%d",&n); a[0] = 1;
    for( i = 2; i <= n; ++i) //外层循环控制阶乘到了第几个数（从2开始）
    {
        for( j = 1, carry = 0; j <= digit; ++j) //内层循环模仿全加器
        {
            temp = a[j-1] * i + carry; //本位乘法之后加上进位
            a[j-1] = temp % 10; //取计算后的本位
            carry = temp / 10; //取计算后的进位
        } while(carry)
        { a[++digit-1] = carry % 10; carry /= 10; //若剩余进位不为0则继续取本位和进位加入a数组
        }
    }
    printf("Result is:\n%d ! = ",n);
    for( i = digit; i >= 1; --i) //倒序输出a数组
    {
        printf("%d",a[i-1]);
    }
    printf("\n");
}
```

代码截图如下:



The screenshot shows the Emacs editor window titled "emacs@localhost.localdomain". The menu bar includes File, Edit, Options, Buffers, Tools, C, and Help. The toolbar contains icons for file operations and editing. The main text area displays a C program that calculates the factorial of a number 'n' using an array 'a'. The program prompts the user to input 'n', then iteratively calculates the factorial by multiplying each digit by 'i' and handling the carry. The result is printed as "Result is:\n%d ! = ", n). Below the code, the status bar shows the file name "hello.c", the current line and column "All L1", and the encoding "(C/*l Abbrev)". At the bottom, there is a sidebar with links to Emacs tutorials and manuals.

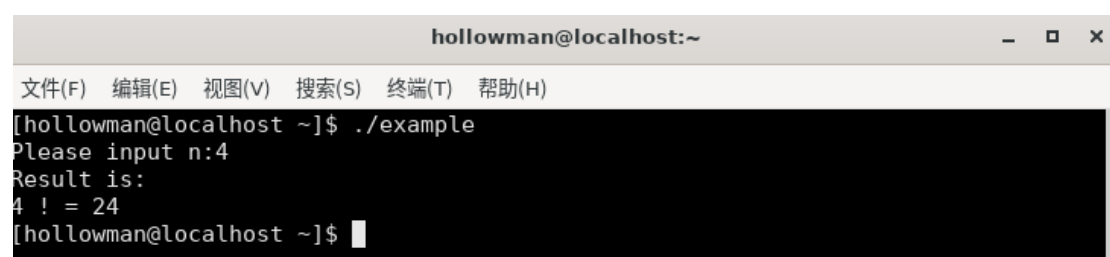
```
#include <stdio.h>
int main()
{
    int n, a[200], carry, temp, i, j, digit = 1;
    printf("Please input n:");
    scanf("%d", &n);
    a[0] = 1;
    for (i = 2; i <= n; ++i)
    {
        for (j = 1, carry = 0; j <= digit; ++j)
        {
            temp = a[j - 1] * i + carry;
            a[j - 1] = temp % 10;
            carry = temp / 10;
        }
        while (carry)
        {
            a[++digit - 1] = carry % 10;
            carry /= 10;
        }
    }
    printf("Result is:\n%d ! = ", n);
    for (i = digit; i >= 1; --i)
    {
        printf("%d", a[i - 1]);
    }
    printf("\n");
}
```

-(DOS)--- hello.c All L1 (C/*l Abbrev)

[Emacs Tutorial](#) Learn basic keystroke commands (Emacs 快速指南)
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info

U:%%- *GNU Emacs* 11% L3 (Fundamental)

求阶乘运行结果如下:



The screenshot shows a terminal window titled "hollowman@localhost:~". The menu bar includes 文件(F), 编辑(E), 视图(V), 搜索(S), 终端(T), and 帮助(H). The terminal output shows the execution of the program: the user enters "4" in response to the prompt "Please input n:", and the program outputs "Result is: 4 ! = 24".

```
[hollowman@localhost ~]$ ./example
Please input n:4
Result is:
4 ! = 24
[hollowman@localhost ~]$
```

3.

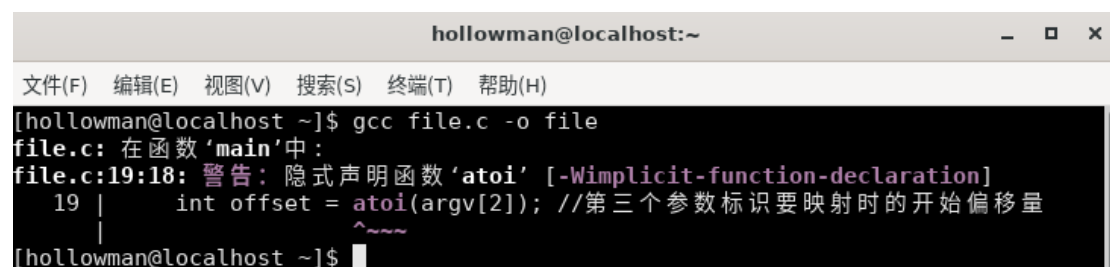
//本程序的功能是把一个文件映射到

//内存然后再把内存中的东西利用write函数给输出到标准输出

//最后利用munmap解除内存映射

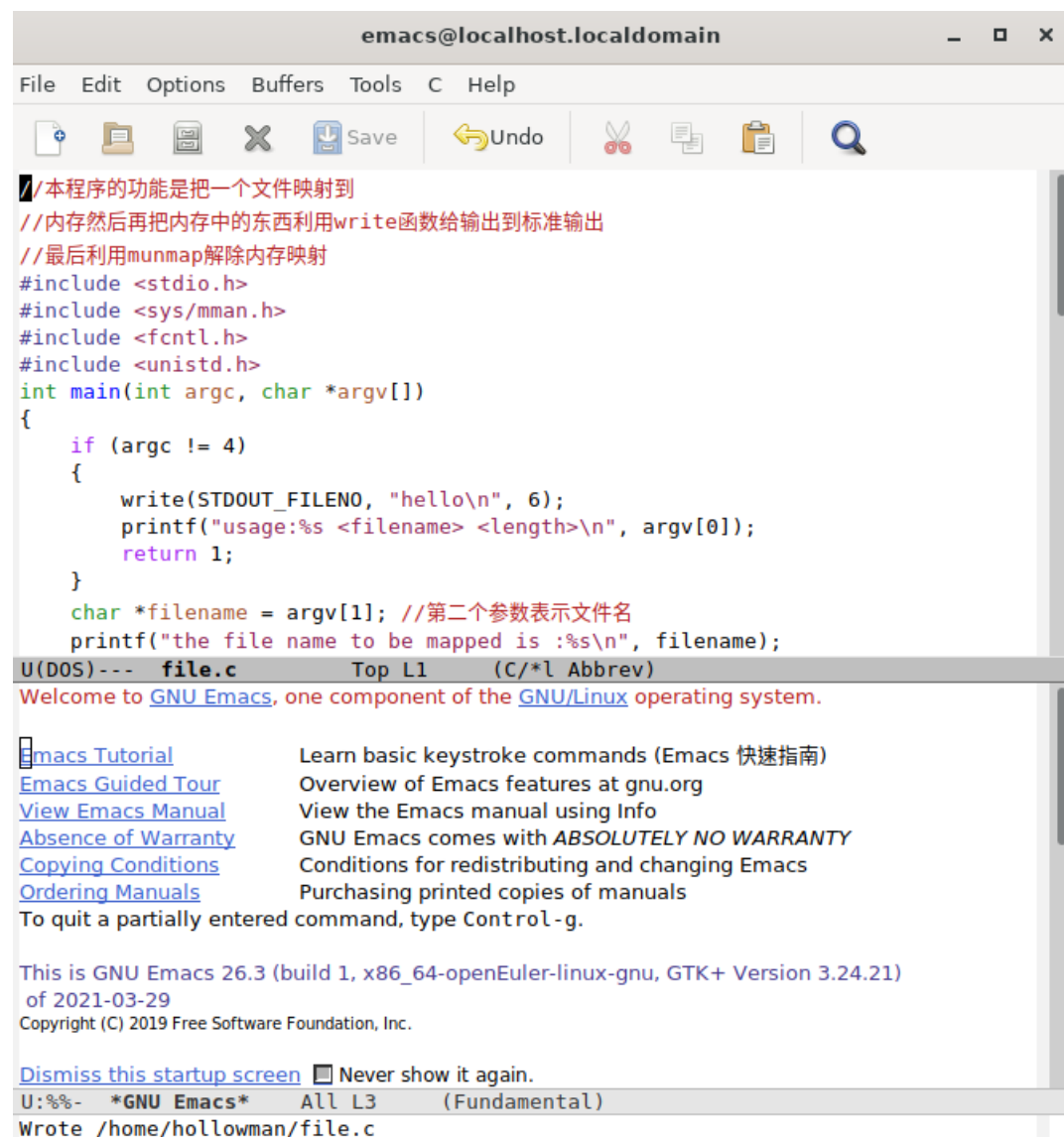
```
#include <stdio.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    if (argc != 4)
    {
        write(STDOUT_FILENO, "hello\n", 6);
        printf("usage:%s <filename> <length>\n", argv[0]);
        return 1;
    }
    char *filename = argv[1]; //第二个参数表示文件名
    printf("the file name to be mapped is :%s\n", filename);
    int fd = open(filename, O_RDONLY);
    int offset = atoi(argv[2]); //第三个参数标识要映射时的开始偏移量
    printf("start offset of file to be mapped is :%d\n", offset);
    printf("page size is %ld\n", sysconf(_SC_PAGE_SIZE));
    int realOffset = offset & ~(sysconf(_SC_PAGE_SIZE) - 1);
    printf("real start offset of file to be mapped is %d\n", realOffset);
    int length = atoi(argv[3]); //要映射的文件的长度
    printf("the length to be map is :%d\n", length);
    int realLen = length + offset - realOffset; //实际映射的长度
    printf("the real length to be map is %d\n", realLen);
    char *addr = mmap(NULL, realLen, PROT_READ, MAP_PRIVATE, fd, realOffset); //有关
    该函数我们可以使用 //man命令去查看
    close(fd);
    write(STDOUT_FILENO, addr, realLen); /*输出到标准输出*/
    munmap(addr, realLen); //解除内存映射
    printf("\n");
    return 0;
}
```

编译结果:



```
hollowman@localhost:~
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
[hollowman@localhost ~]$ gcc file.c -o file
file.c: 在函数 'main' 中:
file.c:19:18: 警告: 隐式声明函数 'atoi' [-Wimplicit-function-declaration]
   19 |     int offset = atoi(argv[2]); //第三个参数标识要映射时的开始偏移量
       |                  ^~~~~
[hollowman@localhost ~]$
```

代码截图：



The screenshot shows the Emacs editor window titled "emacs@localhost.localdomain". The menu bar includes File, Edit, Options, Buffers, Tools, C, and Help. The toolbar contains icons for opening a file, saving, undo, redo, and search. The main text area contains a C program with the following code:

```
//本程序的功能是把一个文件映射到
//内存然后再把内存中的东西利用write函数给输出到标准输出
//最后利用munmap解除内存映射
#include <stdio.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    if (argc != 4)
    {
        write(STDOUT_FILENO, "hello\n", 6);
        printf("usage:%s <filename> <length>\n", argv[0]);
        return 1;
    }
    char *filename = argv[1]; //第二个参数表示文件名
    printf("the file name to be mapped is :%s\n", filename);
}
```

Below the code, the Emacs startup screen is displayed, showing the file name "file.c" and the top of the buffer "L1". The startup screen includes the following text:

Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.

Emacs Tutorial	Learn basic keystroke commands (Emacs 快速指南)
Emacs Guided Tour	Overview of Emacs features at gnu.org
View Emacs Manual	View the Emacs manual using Info
Absence of Warranty	GNU Emacs comes with <i>ABSOLUTELY NO WARRANTY</i>
Copying Conditions	Conditions for redistributing and changing Emacs
Ordering Manuals	Purchasing printed copies of manuals

To quit a partially entered command, type Control-g.

This is GNU Emacs 26.3 (build 1, x86_64-openEuler-linux-gnu, GTK+ Version 3.24.21)
of 2021-03-29
Copyright (C) 2019 Free Software Foundation, Inc.

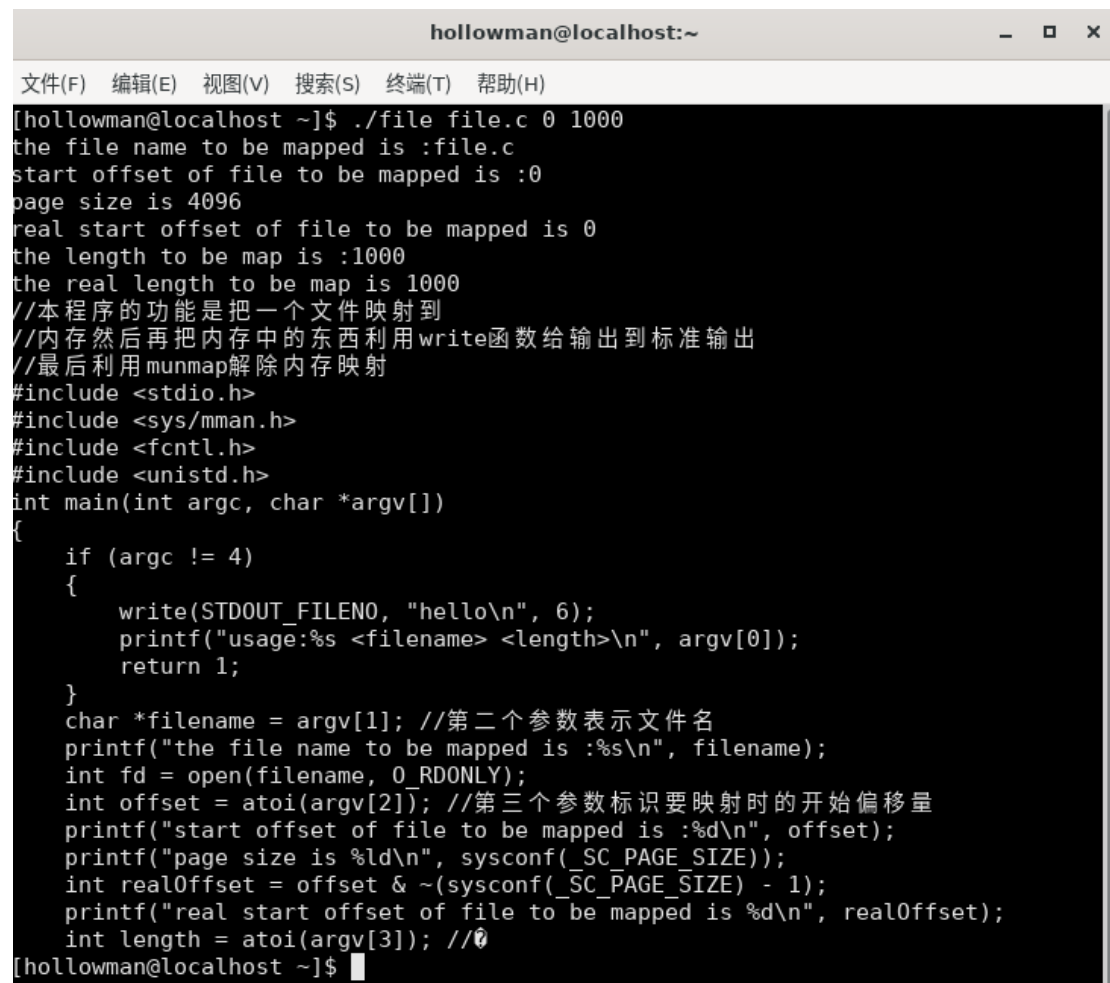
[Dismiss this startup screen](#) ☐ Never show it again.

U:%%- *GNU Emacs* All L3 (Fundamental)

Wrote /home/hollowman/file.c

运行截图：

将源代码文件从开头进行映射并输出，长度为1000.



```
hollowman@localhost:~  
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)  
[hollowman@localhost ~]$ ./file file.c 0 1000  
the file name to be mapped is :file.c  
start offset of file to be mapped is :0  
page size is 4096  
real start offset of file to be mapped is 0  
the length to be map is :1000  
the real length to be map is 1000  
//本程序的功能是把一个文件映射到  
//内存然后再把内存中的东西利用write函数给输出到标准输出  
//最后利用munmap解除内存映射  
#include <stdio.h>  
#include <sys/mman.h>  
#include <fcntl.h>  
#include <unistd.h>  
int main(int argc, char *argv[])  
{  
    if (argc != 4)  
    {  
        write(STDOUT_FILENO, "hello\n", 6);  
        printf("usage:%s <filename> <length>\n", argv[0]);  
        return 1;  
    }  
    char *filename = argv[1]; //第二个参数表示文件名  
    printf("the file name to be mapped is :%s\n", filename);  
    int fd = open(filename, O_RDONLY);  
    int offset = atoi(argv[2]); //第三个参数标识要映射时的开始偏移量  
    printf("start offset of file to be mapped is :%d\n", offset);  
    printf("page size is %ld\n", sysconf(_SC_PAGE_SIZE));  
    int realOffset = offset & ~(sysconf(_SC_PAGE_SIZE) - 1);  
    printf("real start offset of file to be mapped is %d\n", realOffset);  
    int length = atoi(argv[3]); //0  
[hollowman@localhost ~]$
```