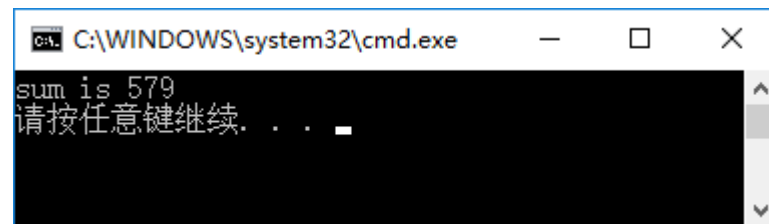


# 最简单的C语言程序

## 【例1.2】求两个整数之和

**解题思路:** 设置3个变量，a和b用来存放两个整数，sum用来存放和数。用赋值运算符“=”把相加的结果传送给sum。



```
C:\WINDOWS\system32\cmd.exe
sum is 579
请按任意键继续. . .
```

```
#include <stdio.h>           //这是编译预处理指令
int main( )                  //定义主函数
{                             //函数开始
    int a,b,sum;             //本行是程序的声明部分，定义a,b,sum为整型变量
    a=123;                   //对变量a赋值
    b=456;                   //对变量b赋值
    sum=a+b;                 //进行a+b的运算，并把结果存放在变量sum中
    printf("sum is %d\n",sum); //输出结果
    return 0;                //使函数返回值为0
}                             //函数结束
```

# 最简单的C语言程序

```
#include <stdio.h>           //这是编译预处理指令
int main( )                  //定义主函数
{                             //函数开始
    int a,b,sum;             //本行是程序的声明部分，定义a,b,sum为整型变量
    a=123;                   //对变量a赋值
    b=456;                   //对变量b赋值
    sum=a+b;                 //进行a+b的运算，并把结果存放在变量sum中
    printf("sum is %d\n",sum); //输出结果
    return 0;                //使函数返回值为0
}                             //函数结束
```

输出时用sum的值取代%d

↓

```
printf("sum is %d\n", sum);
```



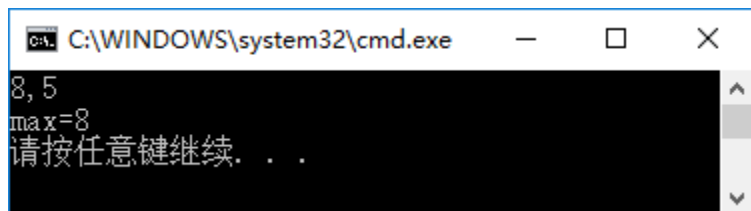
## 程序分析

- **printf("sum is %d\n",sum);** printf函数圆括号内有两个参数。第一个参数是双引号中的内容sum is %d\n，它是输出格式字符串，作用是输出用户希望输出的字符和输出的格式。其中sum is是用户希望输出的字符，**%d**是指定的输出格式，**d**表示用“十进制整数”形式输出。圆括号内第二个参数sum表示要输出变量sum的值。在执行printf函数时，将sum变量的值(以十进制整数表示)取代双引号中的%d。

# 最简单的C语言程序

## 【例1.3】求两个整数中的较大者

**解题思路：** 用一个函数来实现求两个整数中的较大者。在主函数中调用此函数并输出结果。



```
C:\WINDOWS\system32\cmd.exe
8, 5
max=8
请按任意键继续. . .
```

```
#include <stdio.h>
//主函数
int main()
{
    int max(int x,int y); //定义主函数
    int a,b,c; //主函数体开始
    scanf("%d,%d",&a,&b); //对被调用函数max的声明
    c=max(a,b); //定义变量a, b, c
    printf("max=%d\n",c); //输入变量a和b的值
    return 0; //调用max函数, 将得到的值赋给c
} //主函数体结束

//求两个整数中的较大者的max函数
int max(int x,int y) //定义max函数,函数值为整型, 形式参数x和y为整型
{
    int z; //max函数中的声明部分, 定义本函数中用到的变量z为整型
    if(x>y)z=x; //若x>y成立, 将x的值赋给变量z
    else z=y; //否则(即x>y不成立), 将y的值赋给变量z
    return(z); //将z的值作为max函数值, 返回到调用max函数的位置
}
```

# 最简单的C语言程序

```
#include <stdio.h>
//主函数
int main()
{
    int max(int x,int y); //定义主函数
    //主函数体开始
    int a,b,c; //对被调用函数max的声明
    //定义变量a, b, c
    scanf("%d,%d",&a,&b); //输入变量a和b的值
    c=max(a,b); //调用max函数, 将得到的值赋给c
    printf("max=%d\n",c); //输出c的值
    return 0; //返回函数值为0
} //主函数体结束

//求两个整数中的较大者的max函数
int max(int x,int y) //定义max函数,函数值为整型, 形式参数x和y为整型
{
    int z; //max函数中的声明部分, 定义本函数中用到的变量z为整型
    if(x>y)z=x; //若x>y成立, 将x的值赋给变量z
    else z=y; //否则(即x>y不成立), 将y的值赋给变量z
    return(z); //将z的值作为max函数值, 返回到调用max函数的位置
}
```



## 程序分析

- 本程序包括两个函数:①主函数main;  
②被调用的函数max。
- max函数的作用是将x和y中较大者的值赋给变量z, 最后通过return语句将z的值作为max的函数值返回给调用max函数的主函数。
- **scanf**是输入函数的名字(scanf和printf都是C的标准输入输出函数)。该scanf函数的作用是输入变量a和b的值。
- **max(a, b)**调用max函数。在调用时将a和b作为max函数的**实际参数**的值分别传送给max函数中的**形式参数**x和y。

# 最简单的C语言程序

```
#include <stdio.h>
//主函数
int main()
{
    int max(int x,int y); //定义主函数
    //主函数体开始
    int a,b,c; //对被调用函数max的声明
    scanf("%d,%d",&a,&b); //定义变量a, b, c
    c=max(a,b); //输入变量a和b的值
    printf("max=%d\n",c); //调用max函数, 将得到的值赋给c
    return 0; //输出c的值
} //返回函数值为0
//主函数体结束

//求两个整数中的较大者的max函数
int max(int x,int y) //定义max函数,函数值为整型, 形式参数x和y为整型
{
    int z; //max函数中的声明部分, 定义本函数中用到的变量z为整型
    if(x>y)z=x; //若x>y成立, 将x的值赋给变量z
    else z=y; //否则(即x>y不成立), 将y的值赋给变量z
    return(z); //将z的值作为max函数值, 返回到调用max函数的位置
}
```

注意：本例程序中两个函数都有return语句，请注意它们的异同。

两个函数都定义为整型，都有函数值，都需要用return语句为函数指定返回值。

main函数中的return语句指定的返回值一般为0。

max函数的返回值是max函数中求出的两数中的最大值z，只有通过return语句才能把求出的z值作为函数的值并返回调用它的main函数中。

# C语言程序的结构

- 一个程序由一个或多个源程序文件组成
  - 源程序文件包括：预处理指令、全局声明、函数定义
- 函数是C程序的主要组成部分
  - 一个C语言程序是由一个或多个函数组成的，其中必须包含唯一一个main函数
  - 程序中被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己编制设计的函数
- 一个函数包括两个部分：函数首部和函数体，函数体一般包括声明部分和执行部分
- 程序总是从main函数开始执行
- 程序中的操作是由函数中的C语句完成的
- 在每个数据声明和语句的最后必须有一个分号
- C语言本身不提供输入输出语句，输入输出操作由函数完成
- 程序应当包含注释

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

预处理指令

```
int main()
```

main函数

```
{  
    int max(int x,int y);  
    int a,b,c;  
    scanf("%d,%d",&a,&b);  
    c=max(a,b);  
    printf("max=%d\n",c);  
    return 0;  
}
```

库函数调用

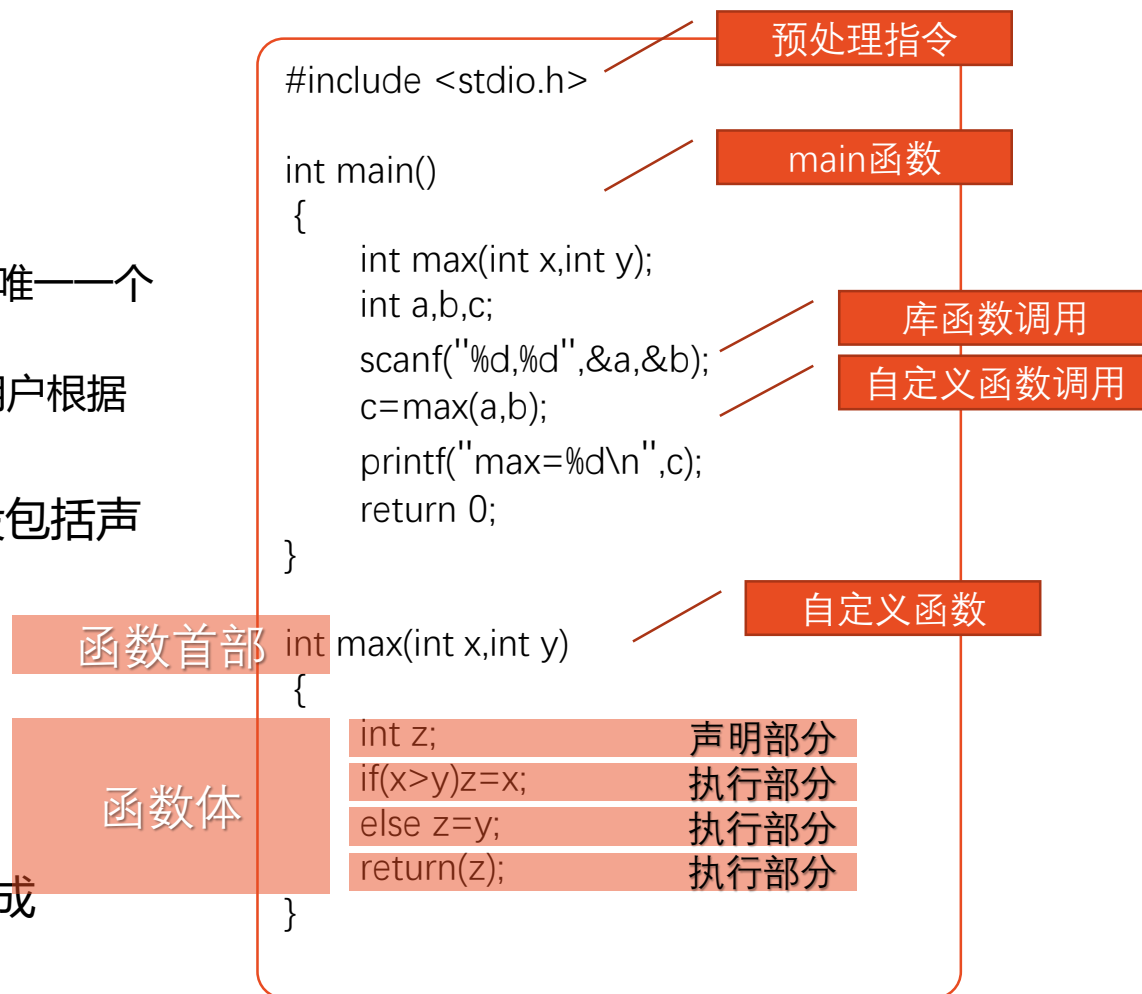
自定义函数调用

```
int max(int x,int y)  
{  
    int z;  
    if(x>y)z=x;  
    else z=y;  
    return(z);  
}
```

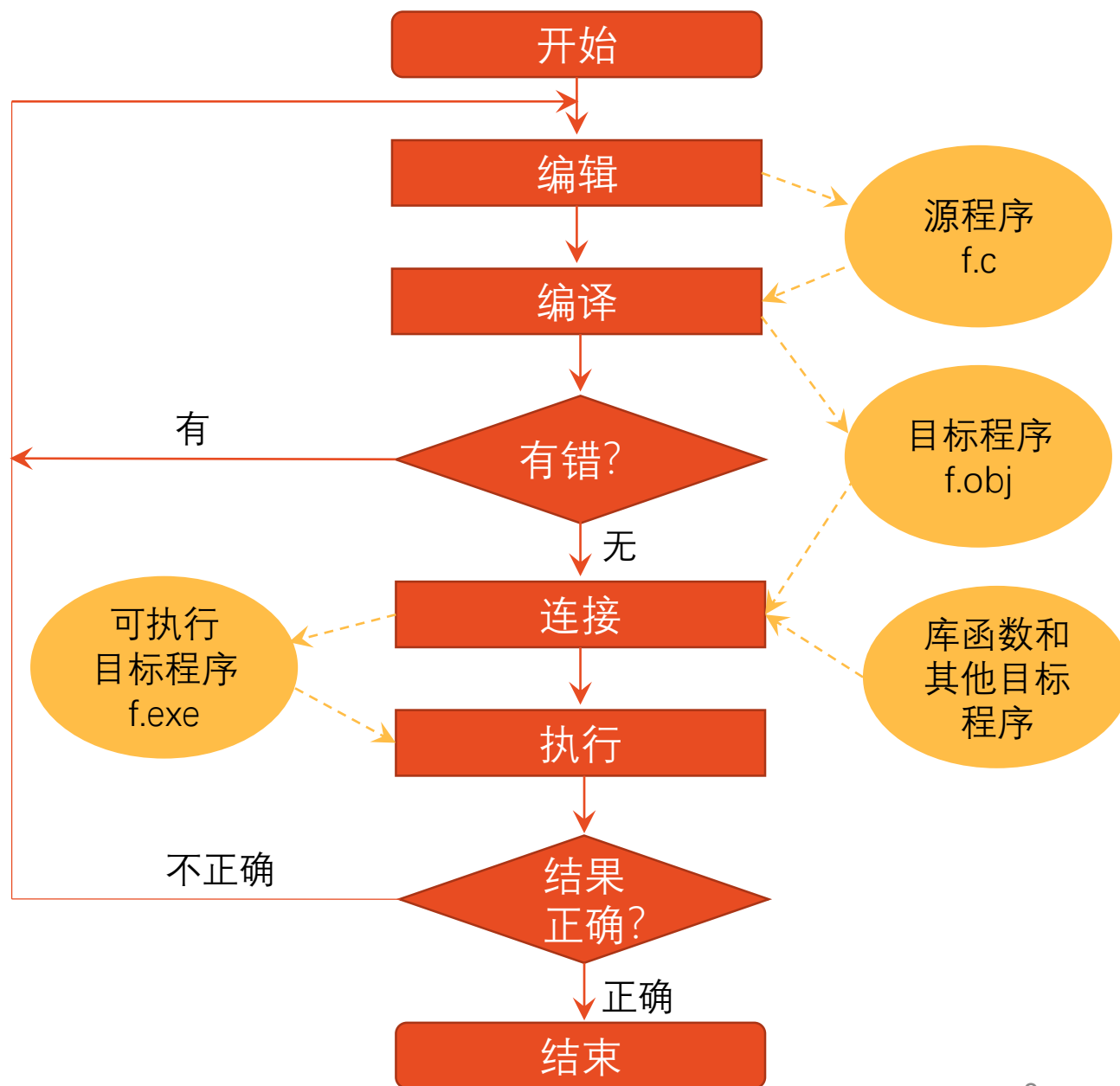
自定义函数

# C语言程序的结构

- 一个程序由一个或多个源程序文件组成
  - 源程序文件包括：预处理指令、全局声明、函数定义
- 函数是C程序的主要组成部分
  - 一个C语言程序是由一个或多个函数组成的，其中必须包含唯一一个main函数
  - 程序中被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己编制设计的函数
- 一个函数包括两个部分：函数首部和函数体，函数体一般包括声明部分和执行部分
- 程序总是从main函数开始执行
- 程序中的操作是由函数中的C语句完成的
- 在每个数据声明和语句的最后必须有一个分号
- C语言本身不提供输入输出语句，输入输出操作由函数完成
- 程序应当包含注释

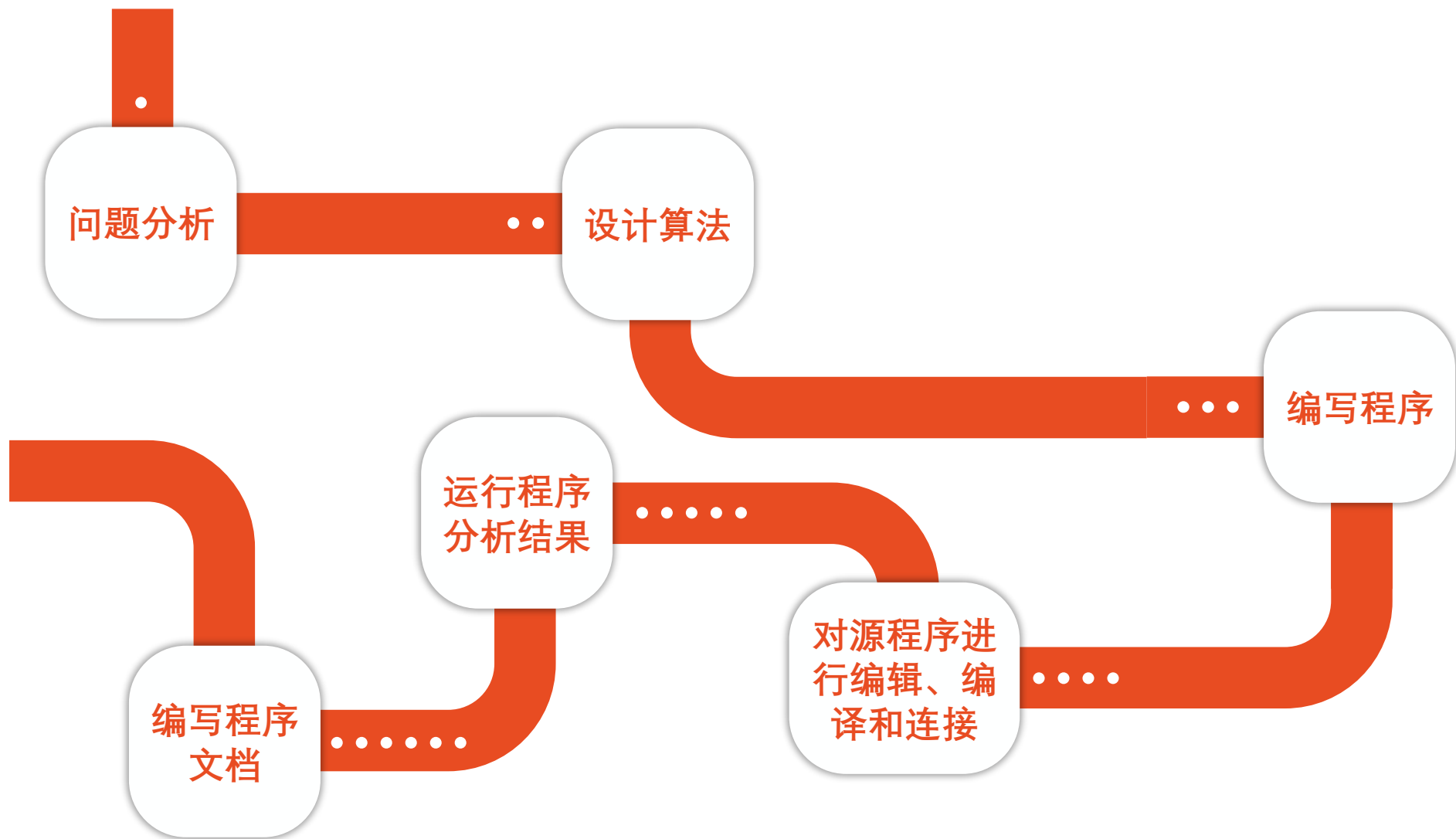


# 运行C程序的步骤与方法





# 程序设计的任务



虚拟机

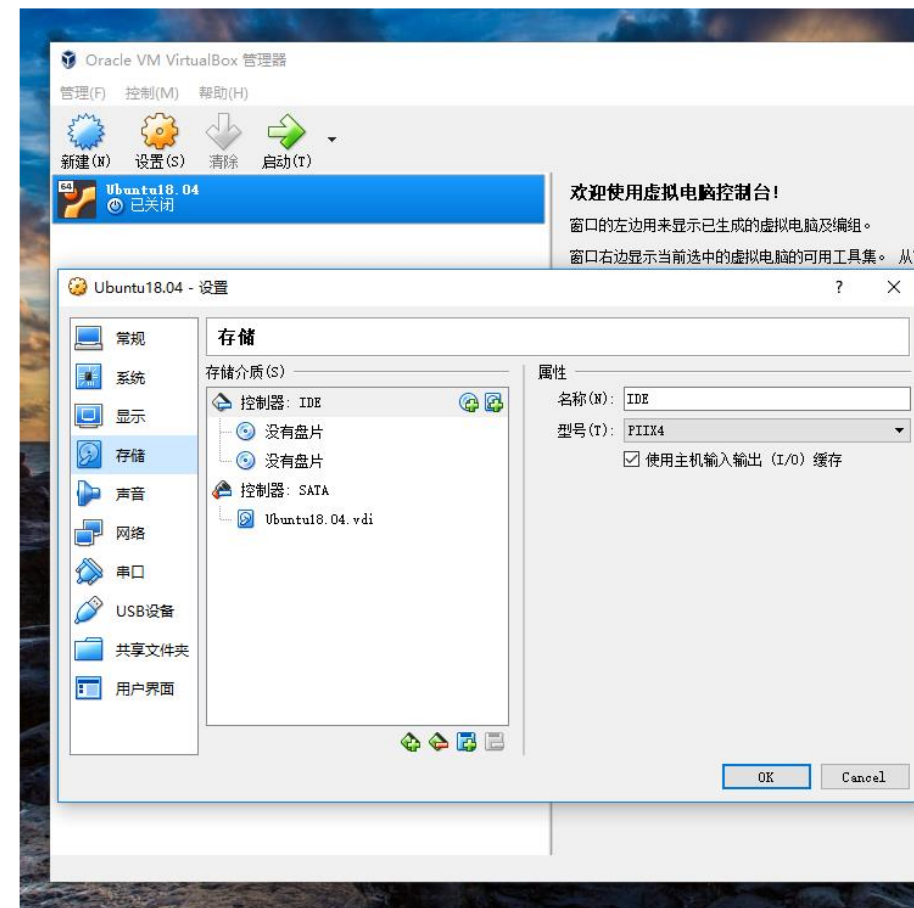
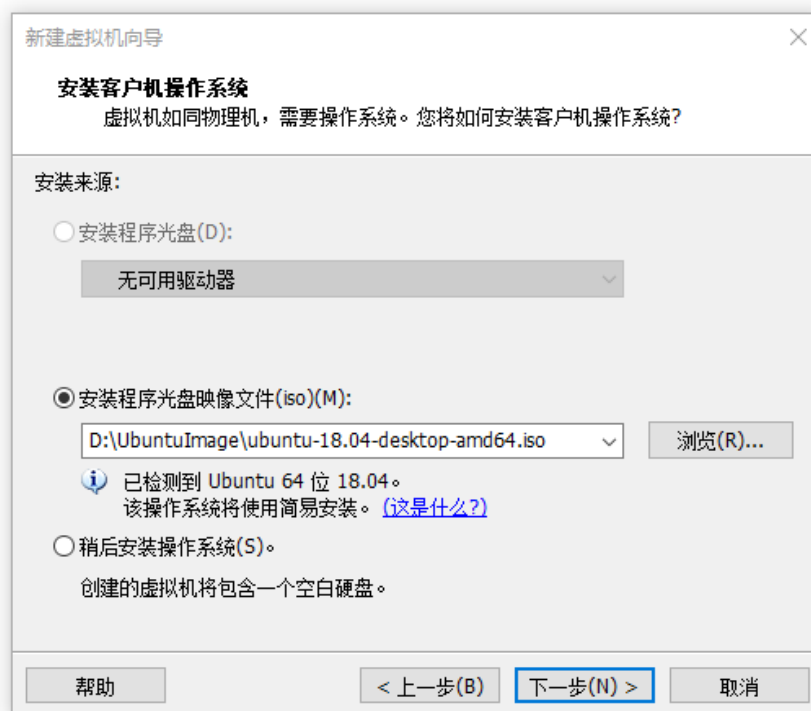
VMWare

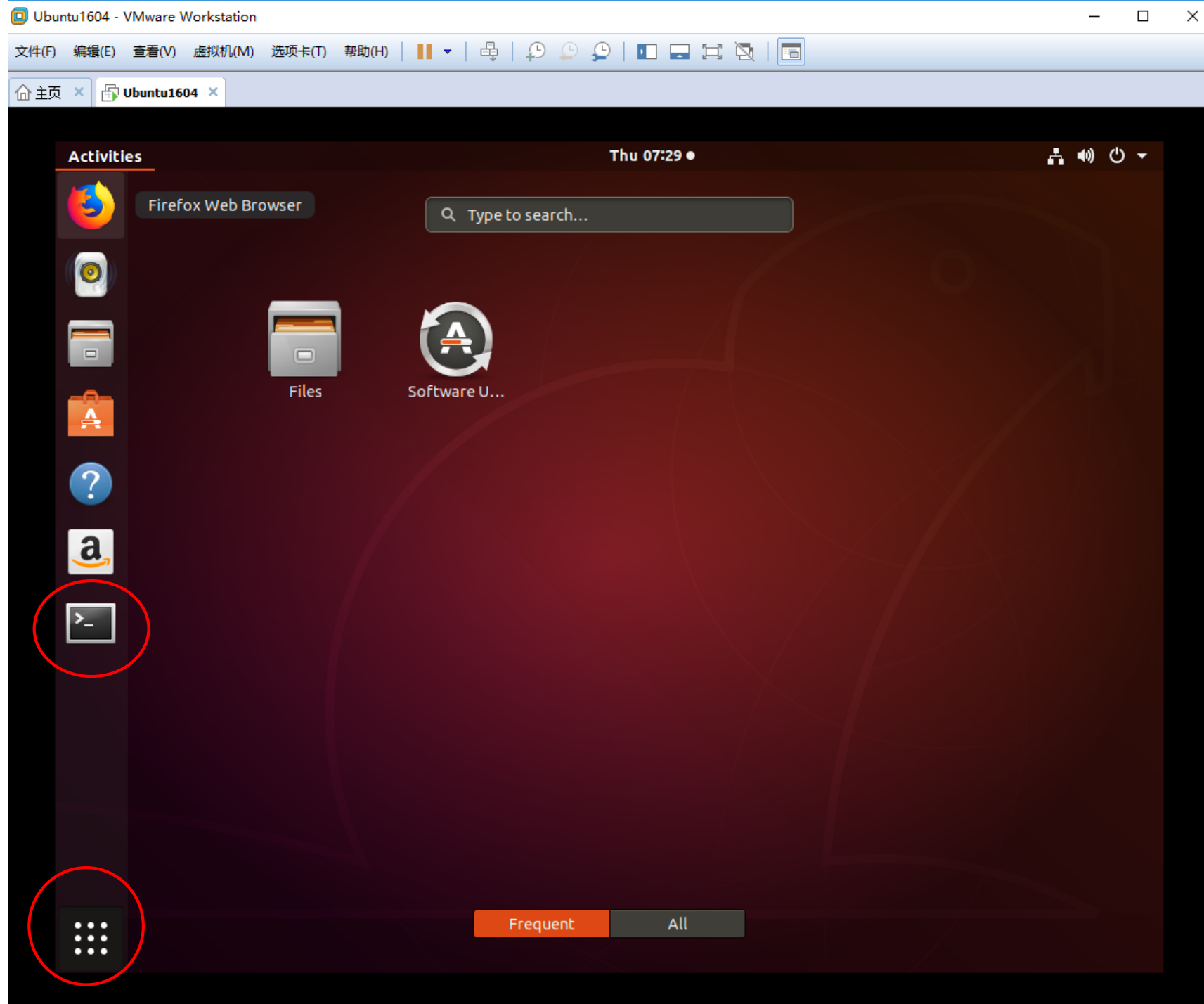
VirtualBox

Linux

Ubuntu (LTS)

## WORKSTATION 14 PRO™







Trash

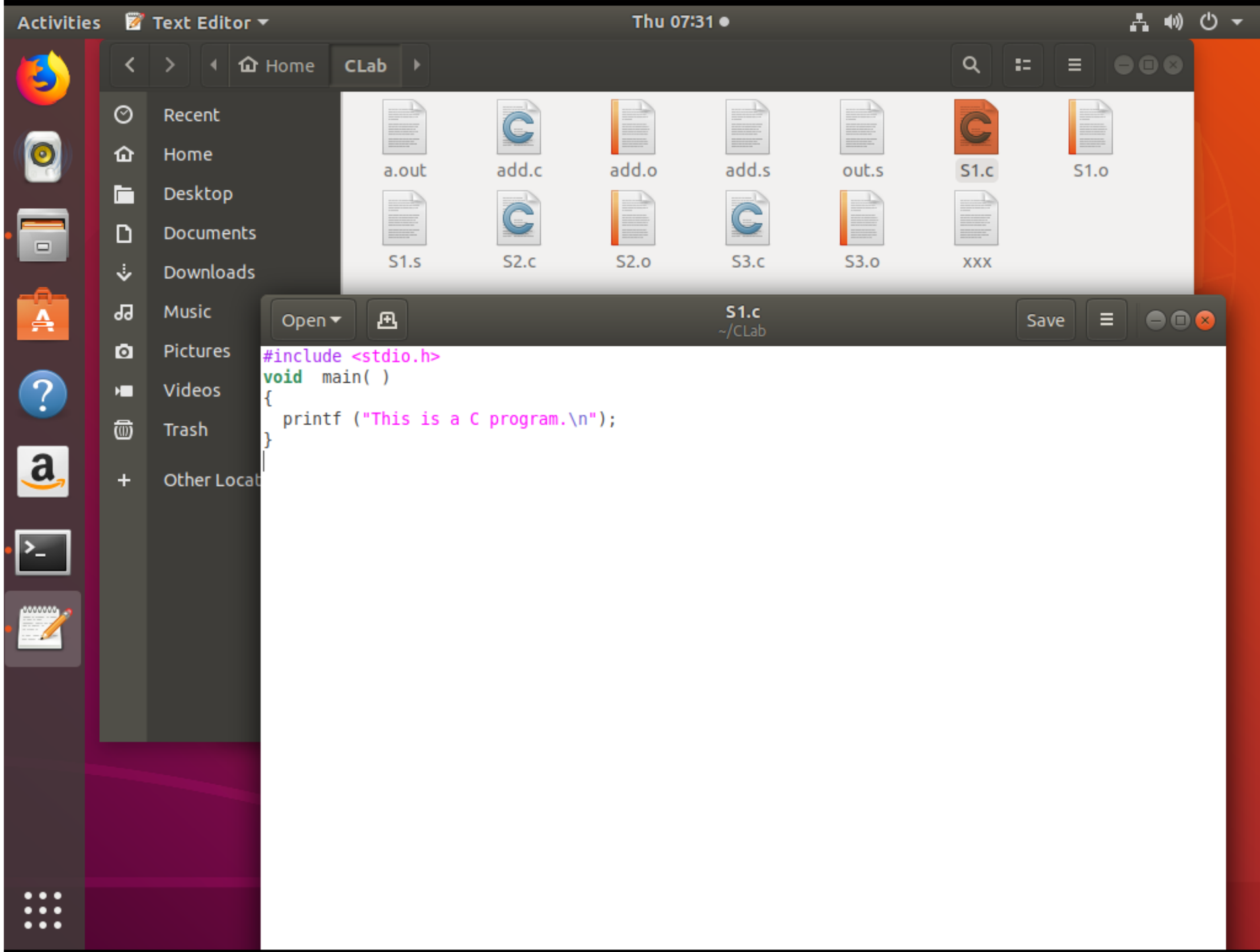


yankaige@ubuntu: ~/CLab

File Edit View Search Terminal Help

```
yankaige@ubuntu:~$ ls
ccaffe-master          gem5
caffe-master.zip       MEVBench
CLab                   Music
core                   neural-networks-and-deep-learning-master
cuda_9.1.85_387.26_linux.run  neural-networks-and-deep-learning-master.zip
DATASet                Pictures
DeepLearning           Projects
Desktop                Public
DNNLab                 PyTorchCode
DNNLab0403.zip         SCALE-Sim-master.zip
Documents              SCALE-Sim-test
Downloads              Templates
examples.desktop       Videos
faster-rcnn.pytorch

yankaige@ubuntu:~$ cd CLab/
yankaige@ubuntu:~/CLab$ ls
add.c  add.o  add.s  a.out  out.s  S1.c  S1.o  S1.s  S2.c  S2.o  S3.c  S3.o
yankaige@ubuntu:~/CLab$ touch xxx
yankaige@ubuntu:~/CLab$ ls
add.c  add.s  out.s  S1.o  S2.c  S3.c  xxx
add.o  a.out  S1.c  S1.s  S2.o  S3.o
yankaige@ubuntu:~/CLab$
```





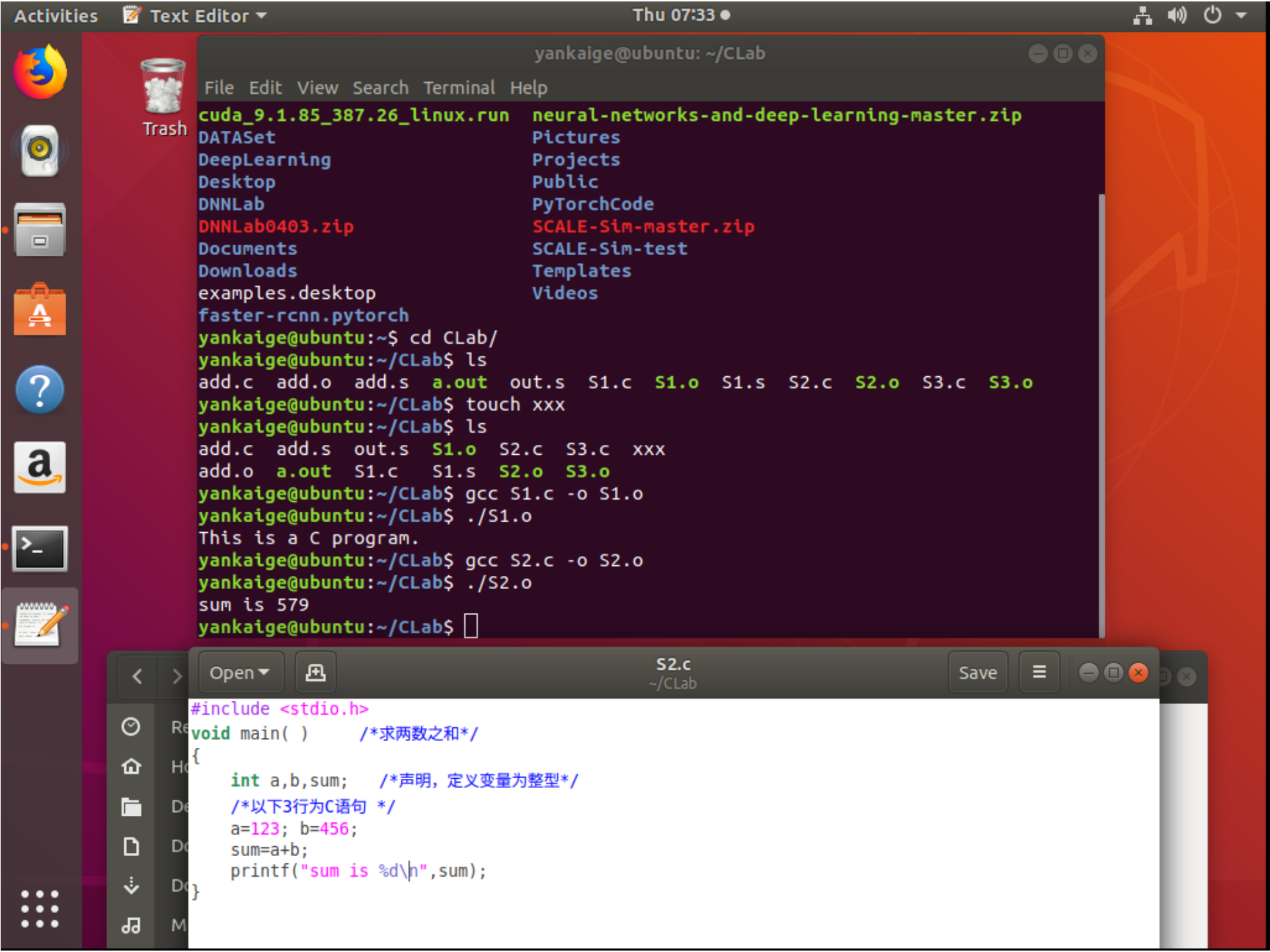
Trash



Show Applications

```
yankaige@ubuntu: ~/CLab
File Edit View Search Terminal Help
caffe-master.zip      MEVBench
CLab                  Music
core                  neural-networks-and-deep-learning-master
cuda_9.1.85_387.26_linux.run  neural-networks-and-deep-learning-master.zip
DATASet              Pictures
DeepLearning         Projects
Desktop              Public
DNNLab               PyTorchCode
DNNLab0403.zip       SCALE-Sim-master.zip
Documents            SCALE-Sim-test
Downloads            Templates
examples.desktop     Videos
faster-rcnn.pytorch
yankaige@ubuntu:~$ cd CLab/
yankaige@ubuntu:~/CLab$ ls
add.c  add.o  add.s  a.out  out.s  S1.c  S1.o  S1.s  S2.c  S2.o  S3.c  S3.o
yankaige@ubuntu:~/CLab$ touch xxx
yankaige@ubuntu:~/CLab$ ls
add.c  add.s  out.s  S1.o  S2.c  S3.c  xxx
add.o  a.out  S1.c  S1.s  S2.o  S3.o
yankaige@ubuntu:~/CLab$ gcc S1.c -o S1.o
yankaige@ubuntu:~/CLab$ ./S1.o
This is a C program.
yankaige@ubuntu:~/CLab$
```

```
Open S1.c ~/CLab Save
#include <stdio.h>
void main( )
{
    printf ("This is a C program.\n");
}
```





Activities Text Editor Thu 07:34

yankaige@ubuntu: ~/CLab

File Edit View Search Terminal Help

DNNLab PyTorchCode  
DNNLab0403.zip SCALE-Sim-master.zip  
Documents SCALE-Sim-test  
Downloads Templates  
examples.desktop Videos  
faster-rcnn.pytorch

```
yankaige@ubuntu:~$ cd CLab/  
yankaige@ubuntu:~/CLab$ ls  
add.c add.o add.s a.out out.s S1.c S1.o S1.s S2.c S2.o S3.c S3.o  
yankaige@ubuntu:~/CLab$ touch xxx  
yankaige@ubuntu:~/CLab$ ls  
add.c add.s out.s S1.o S2.c S3.c xxx  
add.o a.out S1.c S1.s S2.o S3.o  
yankaige@ubuntu:~/CLab$ gcc S1.c -o S1.o  
yankaige@ubuntu:~/CLab$ ./S1.o  
This is a C program.  
yankaige@ubuntu:~/CLab$ gcc S2.c -o S2.o  
yankaige@ubuntu:~/CLab$ ./S2.o  
sum is 579  
yankaige@ubuntu:~/CLab$ gcc S3.c -o S3.o  
yankaige@ubuntu:~/CLab$ ./S3.o  
4,10  
max=10  
yankaige@ubuntu:~/CLab$
```

Open S3.c ~/CLab

```
#include <stdio.h>  
void main( ) /* 主函数*/  
{  
    int max(int x,int y); /* 对被调用函数max的声明 */  
    int a, b, c; /*定义变量a、b、c */  
    scanf("%d,%d",&a,&b); /*输入变量a和b的值*/  
    c=max(a,b); /*调用max函数,将得到的值赋给c */  
    printf("max=%d\n",c); /*输出c的值*/  
}  
  
int max(int x, int y)  
{  
    int z;  
    if (x>y) z=x;  
    else z=y;  
    return (z);  
}
```

Recent Home Desktop Documents Downloads Music

a.out add.c add.o S1.s S2.c S2.o