Getconf LONG\_BIT

获取系统位数

============================函数（模块化）

一、为什么要有函数？

一台电脑 = 鼠标 + 键盘 + 显示器 + 主机

高内聚——————》模块内部紧凑 严谨

低耦合——————》不同模块之间的联系

二、函数怎么写？

格式

返回值类型 函数名（参数列表） //-->这一行是函数头

{

函数体

}

模板

1、无返回值无参数

|  |  |
| --- | --- |
| Void | function（void）//参数里面的void 可以省略 |

2、无返回值有参数

|  |  |
| --- | --- |
| Void | function（int a，char b） |

3、有返回值无参数

|  |  |
| --- | --- |
| Int | function（void） |
| {  //…  Return （？）；  } |  |

？的类型需要和返回值类型一致，括号可有可无，一旦执行 return ，函数就结束了。

4、有返回值有参数

|  |  |
| --- | --- |
| Int | function（int a，int b） |
| {  //…  Return （？）；  } |  |

三、函数应该注意什么

1、返回值类型可有可无，如果为无就写 void

2、可以有多个参数，每个参数之间使用逗号隔开，参数可有可无，如果为无可以写void也可以不写

形参

-------》形式上的参数，其实就是喊口号的

实参

-------》实际的参数，可有传变量、常量以及表达式

参数传递分为以下两种：

值传递

实参传给形参，单向传输

地址传递

传地址

3、函数名必须符合标识符命名规范，而且一定要见名知意

4、函数有 定义、声明、调用

5、声明时，变量名可省略

6、函数名就是函数的入口地址

现场保护？

在跳到相应函数执行之前需要记住回去的路

现场恢复？

你执行函数结束之后是不是还要回去

使用了一个特殊寄存器叫做链接寄存器（LR）

这个寄存器用于保存返回地址

四、除了自定义函数还有一些是别人写好的

1、输入输出函数（以f开头的都是标准库函数）

Printf

Fprintf

Sprintf

Scanf

Sscanf

Fscanf

Get 家族

头文件

#include <stdio.h>

函数原型

int fgetc(FILE \*stream); ---------》获取一个字符，返回你获取的字符

char \*fgets(char \*s, int size, FILE \*stream);

int getc(FILE \*stream);

int getchar(void); ----------》默认从键盘获取一个字符

char \*gets(char \*s); ---------》默认从键盘获取一串字符串

int ungetc(int c, FILE \*stream);

参数

Stream----》字节流，一般写stdin，表示从键盘获取数据

Stdout ----》标准输出，指的就是显示器

功能

获取字符或字符串

put家族

#include <stdio.h>

int fputc(int c, FILE \*stream);

int fputs(const char \*s, FILE \*stream);

int putc(int c, FILE \*stream);

int putchar(int c);

int puts(const char \*s);

字符串相关函数

#include <string.h>

char \*strcpy(char \*dest, const char \*src);

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字符串复制，从哪复制到哪？ |  |  | src源 | dest目标 |

char \*strncpy(char \*dest, const char \*src, size\_t n);

NAME

strcmp, strncmp - compare two strings

SYNOPSIS

#include <string.h>

int strcmp(const char \*s1, const char \*s2);

int strncmp(const char \*s1, const char \*s2, size\_t n);

NAME

strcat, strncat - concatenate two strings

SYNOPSIS

#include <string.h>

char \*strcat(char \*dest, const char \*src);

char \*strncat(char \*dest, const char \*src, size\_t n);