西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计

Programming in C

1011014

主讲：姜学锋，计算机学院

学会用指针数据

1、函数返回指针数据

2、函数指针

3、带参数的main函数

7.5.2 函数返回指针值

- 函数的返回类型可以是指针类型，即函数返回指针值，其定义形式为：

```
返回类型 *函数名(形式参数列表)
{
    函数体
}
```

7.5.2 函数返回指针值

- 例如：

```
char* substring(const char *str, const char *sub)
{
   ...//函数体
}
```

7.5.2 函数返回指针值

- 函数返回指针值，需要考虑指针有效性的问题，例如：

```
char* substring(const char *str, const char *sub)
{
   char a='A';
   return &a; //正确 返回值&a与返回类型char *匹配
}
```

- 这个返回有问题，因为它返回的是函数局部变量a的地址值。当函数调用结束后，函数局部变量会释放，变成未知对象。在return语句时，&a还是有效的，但主调函数获得这个地址时已经是无效的。

7.5.2 函数返回指针值

- 一般地，函数应返回：
- ①由主调函数传递进去的有效指针值；
- ②由动态分配得到的指针值（后面将要讲到）；
- ③0值指针，表示无效指针。

7.5.2 函数返回指针值

【例7.22】

编写函数stringstr，实现strstr函数的查找子字符串功能。

7.5.2 函数返回指针值

例7.22

```
 1 #include <stdio.h>
 2 const char *stringstr(const char *string, const char *strC
harSet)
 3 {
 4   const char* p=string, *r=strCharSet;
 5   while(*p!='\0') {
 6     while( *p++ == *r++) ; //比较直到字符串结束或不相等为止
 7     if(*r=='\0') return p; //包含strCharSet返回string当前指针
 8     r=strCharSet; //重新指向strCharSet
 9     p=++string; //从string下一个字符起始
10   }
11   return NULL; //不包含strCharSet返回NULL
12 }
13 int main()
14 {
```

7.5.2 函数返回指针值

例7.22

```
15    char s1[80]="*A*AB*ABC*ABCD", s2[80]="ABC";
16    const char *ptr;
17    ptr=(char *)stringstr(s1,s2);
18    if (ptr!=NULL) printf("%s\n",ptr);
19    return 0;
20 }
```

程序运行结果如下：

ABC*ABCD

7.5.2 函数返回指针值

```
 2 const char *stringstr(const char *string, const char *strC
harSet)
 3 {
 4   const char* p=string, *r=strCharSet;
 5   while(*p!='\0') {
 6     while( *p++ == *r++) ; //比较直到字符串结束或不相等为止
 7     if(*r=='\0') return p; //包含strCharSet返回string当前指针
 8     r=strCharSet; //重新指向strCharSet
 9     p=++string; //从string下一个字符起始
10   }
11   return NULL; //不包含strCharSet返回NULL
12 }
```

stringstr函数的作用是在string字符串中查找有无与strCharSet相同的字符串，如果有，返回该字符串在string中的位置的指针，否则返回空指针表示没有相同的字符串

7.5.2 函数返回指针值

```
 2 const char *stringstr(const char *string, const char *strC
harSet)
 3 {
 4   const char* p=string, *r=strCharSet;
 5   while(*p!='\0') {
 6     while( *p++ == *r++) ; //比较直到字符串结束或不相等为止
 7     if(*r=='\0') return p; //包含strCharSet返回string当前指针
 8     r=strCharSet; //重新指向strCharSet
 9     p=++string; //从string下一个字符起始
10   }
11   return NULL; //不包含strCharSet返回NULL
12 }
```

程序第6行是字符串比较的关键，无论p或是r指向的字符串，只要指向的字符串有不相同的字符，循环就结束。此时有3中情况：

①p和r均没有指向两个字符串的结束，说明字符串中间就有字符不相等

7.5.2 函数返回指针值

```
 2 const char *stringstr(const char *string, const char *strC
harSet)
 3 {
 4   const char* p=string, *r=strCharSet;
 5   while(*p!='\0') {
 6     while( *p++ == *r++) ; //比较直到字符串结束或不相等为止
 7     if(*r=='\0') return p; //包含strCharSet返回string当前指针
 8     r=strCharSet; //重新指向strCharSet
 9     p=++string; //从string下一个字符起始
10   }
11   return NULL; //不包含strCharSet返回NULL
12 }
```

此时有3中情况：

①p和r均没有指向两个字符串的结束，说明字符串中间就有字符不相等

7.5.2 函数返回指针值

```
const char *stringstr(const char *string, const char *strCharSet)
{
  const char* p=string, *r=strCharSet;
  while(*p!='\0') {
    while( *p++ == *r++) ; //比较直到字符串结束或不相等为止
    if(*r=='\0') return p; //包含strCharSet返回string当前指针
    r=strCharSet; //重新指向strCharSet
    p=++string; //从string下一个字符起始
  }
  return NULL; //不包含strCharSet返回NULL
}
```

②p指向字符串结束，r没有指向字符串的结束，说明r后面还有没有比较的字符；

7.5.2 函数返回指针值

```
 2 const char *stringstr(const char *string, const char *strC
harSet)
 3 {
 4   const char* p=string, *r=strCharSet;
 5   while(*p!='\0') {
 6     while( *p++ == *r++) ; //比较直到字符串结束或不相等为止
 7     if(*r=='\0') return p; //包含strCharSet返回string当前指针
 8     r=strCharSet; //重新指向strCharSet
 9     p=++string; //从string下一个字符起始
10   }
11   return NULL; //不包含strCharSet返回NULL
12 }
```

③p尚未指向字符串结束，r指向字符串的结束；显然，第3种情况说明p所指向的字符串包含了strCharSet字符串，则r应指向结束符。

7.5.3 函数指针

- 函数是实现特定功能的程序代码的集合，实际上，函数代码在内存中也要占据一段存储空间（代码区内），这段存储空间的起始地址称为函数入口地址。C语言规定函数入口地址为函数的指针，即函数名既代表函数，又是函数的指针（或地址）。

7.5.3 函数指针

- C语言允许定义指向函数的指针变量，定义形式为：

```
返回类型 (*函数指针变量名)(形式参数列表), ......;
```

- 它可以指向形如

```
返回类型 函数名(形式参数列表)
{
    函数体
}
```

- 函数。

7.5.3 函数指针

- 需要注意定义形式中的括号不能省略。例如：

```
int (*p)(int a, int b); //定义函数指针变量
```

7.5.3 函数指针

- 与数据对象指针不同，函数指针一般只有赋值和间接引用的操作，其他运算不适用。

7.5.3 函数指针

- 1．指向函数
- 可以将函数的地址赋值给函数指针变量，形式为

```
函数指针变量=函数名;
```

- 它要求函数指针变量与指向函数必须有相同的返回类型、参数个数、参数类型（即函数原型相同）。

7.5.3 函数指针

- 例如假设：

```
int max(int a, int b); //max函数原型
int min(int a, int b); //min函数原型
int (*p)(int a, int b); //定义函数指针变量
```

- 则

```
p=max;
```

- 称p指向函数max。它也可以指向函数min，即可以指向所有与它有相同函数原型的函数。

7.5.3 函数指针

- 2．通过函数指针调用函数
- 对函数指针间接引用即是通过函数指针调用函数，一般形式为：

```
①(*函数指针)(实参列表)
②函数指针(实参列表)
```

- 两种形式是完全相同的。通常，程序员偏爱用第②种形式。

7.5.3 函数指针

- 通过函数指针调用函数，在实参、参数传递、返回值等方面与函数名调用相同。例如：

```
c=p(a,b); //等价于c=max(a,b);
```

7.5.3 函数指针

【例7.23】

通过函数指针调用max和min函数。

7.5.3 函数指针

例7.23

```
 1 #include <stdio.h>
 2 int max(int a, int b) //求最大值
 3 {
 4   return a>b ? a:b ;
 5 }
 6 int min(int a, int b) //求最小值
 7 {
 8   return a<b ? a:b ;
 9 }
10 int main()
11 {
12   int (*p)(int a, int b); //定义函数指针变量
13   p=max; //p指向max函数
14   printf("%d ",p(3,4)); //通过p调用函数
15   p=min; //p指向min函数
```

7.5.3 函数指针

例7.23

```
16    printf("%d ",p(3,4)); //通过p调用函数
17    return 0;
18 }
```

7.5.3 函数指针

```
10 int main()
11 {
12   int (*p)(int a, int b); //定义函数指针变量
13   p=max; //p指向max函数
14   printf("%d ",p(3,4)); //通过p调用函数
15   p=min; //p指向min函数
16   printf("%d ",p(3,4)); //通过p调用函数
17   return 0;
18 }
```

从中看出，函数调用p(3,4)究竟调用max或者min，取决于调用前p指向哪个函数。

7.5.3 函数指针

- 3．函数指针的用途
- 指向函数的指针多用于指向不同的函数，从而可以利用指针变量调用不同函数，相当于将函数调用由静态方式（固定地调用指定函数）变为动态方式（调用哪个函数是由指针值来确定）。熟练掌握函数指针的应用，有利于程序的模块化设计，提高程序的可扩展性。

7.5.3 函数指针

- 实际编程中，函数指针在菜单设计、事件驱动、动态链接库等领域得到充分的应用。

7.7 带参数的main函数

- C语言标准中的main函数允许带有参数，定义形式为：

```
int main(int argc, char *argv[])
{
   ......//函数体
}
```

- 其中，第1个参数argc表示命令行中字符串的个数，是非负整数值。第2个参数argv是一个字符串指针数组，用于指向命令行中各个字符串。

7.7 带参数的main函数

- 需要注意，argv[argc]是一个空指针。如果argc大于1，则argv[0]是一个指向程序名的字符串指针，argv[1]～argv[argc-1]是指向命令行参数的字符串指针。换言之，通过argv[0]可以得到程序名称，通过argv[1]～argv[argc-1]可以得到命令行参数。

7.7 带参数的main函数

- 一个命令行程序在系统提示符中是按如下格式的命令输入的：

```
可执行程序名  参数1    参数2    参数3  ......
```

- 其中用空格作为间隔。

7.7 带参数的main函数

- 按上述命令形式执行时，系统会将命令行各个参数传递到main函数中，通过argc和argv两个参数可以让程序得到命令行上的信息，具体为：
- ①argc：命令行中字符串的个数（含可执行程序名称）；
- ②argv[0]：可执行程序名称字符串的首地址；
- ③argv[1]：参数1字符串的首地址；
- ④argv[2]：参数2字符串的首地址，其余以此类推。

7.7 带参数的main函数

【例7.27】

编写程序输出命令行信息。

7.7 带参数的main函数

例7.27

```
1 #include <stdio.h>
2 int main(int argc, char *argv[])
3 {
4   int i;
5   if (argc>0) {
6     printf("program:%s ",argv[0]); //输出程序名
7     for (i=1; i<argc; i++)
8       printf("%s ",argv[i]); //输出程序参数字符串
9   }
10  return 0;
11 }
```

7.7 带参数的main函数

- 假定程序取名TEST，在命令行提示符中输入以下命令
- C:\>TEST /i /u /h /? IN.DAT OUT.DAT

- 程序运行结果如下：
- program:TEST /i /u /h /? IN.DAT OUT.DAT


程序设计