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

大批量数据的简洁表示

3、多级指针

7.3.5 指向指针的指针

- 作为指针变量的内存单元也有地址。显然，存放指针变量地址的变量还是一个指针变量，是指向指针类型的指针变量，称为指向指针的指针，即二级指针变量（二级指针）。其定义形式为：

```
指针类型 **指针变量名，……
```

- C语言允许定义多级指针变量，一般形式为：

```
指针类型 *……*指针变量名，……
```

7.3.5 指向指针的指针

- 例如：

```
int **pp; //定义二级指针变量
```

- 表示指针变量pp是一个指向整型指针的指针变量。

7.3.5 指向指针的指针

- 假定，非指针对象为普通对象，称前面述及的指针变量为一级指针变量。
- 普通对象、一级指针变量、二级指针变量可以同时定义，例如：

```
int  a, *p, **pp; //定义普通对象，一级指针变量，二级指针变量
```

7.3.5 指向指针的指针

- 二级指针变量或多级指针变量的使用、运算等操作与一级指针变量相同。如果没有对二级指针变量初始化，则该变量存放的地址是无效的，这时不能使用它做间接引用。

7.3.5 指向指针的指针

- 假设已知

```
int a=20, *p=&a, **pp=&p;
```

- 指针变量p的初值为整型变量a的地址，指针变量pp的初值为指针变量p的地址。则有：
  - （1）*p：间接引用运算的结果为a；
  - （2）*pp：间接引用运算的结果为p；
  - （3）**pp：等价于*(*pp)，两次间接引用运算的结果为a；
  - （4）pp变量的值是指针变量p的地址；
  - （5）p变量的值是整型变量a的地址。

7.3.5 指向指针的指针

图7.16 指针的指针的含义

7.3.5 指向指针的指针

【例7.13】

输出一级指针变量、二级指针变量的地址值和间接引用值。

7.3.5 指向指针的指针

例7.13

```
1 #include <stdio.h>
2 int main()
3 {
4   int a=20, *p=&a, **pp=&p;
5   printf("a=%d\t*p=%d\t**pp=%d\n",a,*p,**pp);
6   printf("&a=%x\tp=%x\t*pp=%x\n",&a,p,*pp);
7   printf("&p=%x\tpp=%x\t&pp=%x\n",&p,pp,&pp);
8   return 0;
9 }
```

程序某次运行结果如下：

```
a=20          *p=20          **pp=20
&a=12ff7c     p=12ff7c       *pp=12ff7c
&p=12ff78     pp=12ff78      &pp=12ff74
```

7.3.5 指向指针的指针

- 为指针变量赋值或初始化时，指针级别不能混淆，即一级指针变量只能取得普通对象的地址，二级指针变量只能取得一级指针变量的地址，以此类推，否则指向类型不一致。例如：

```
int a=10, *p=&a, **pp=&a; //pp=&a错误
pp=&a; //错误
pp=p; //错误
```

- pp的指向类型为“int *”，而&a和p的指向类型为“int”。

7.3.5 指向指针的指针

- 实际编程中，指向指针的指针通常和指针数组结合在一起使用，一般用作函数参数。

7.3.5 指向指针的指针

- 假设已知

```
int a[4][4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};//二维数组
int *s[4]={a[0],a[1],a[2],a[3]}; //一维指针数组初始化
int **pp=s; //二级指针指向一维指针数组
```

- 则通过二级指针访问一维指针数组s的典型形式为：
  - ①pp=s、pp=&s[0]：二级指针pp指向一维指针数组s的首元素地址；
  - ②pp=s+i、pp=&s[i]、pp+i：二级指针pp指向一维指针数组s[i]地址；
  - ③*(pp+i)、pp[i]：等价于s[i]。

7.3.5 指向指针的指针

- 通过二级指针访问一维指针数组s再间接访问二维数组a的典型形式为：
- ①*(pp+i)、pp[i]：指向a[i][0]；
- ②**(pp+i)、*pp[i]、*(pp[i]+0)、pp[i][0]：等价于a[i][0]；
- ③*(pp+i)+j、pp[i]+j：指向a[i][j]；
- ④*(*(pp+i)+j)、*(pp[i]+j)、pp[i][j]：等价于a[i][j]；

7.3.5 指向指针的指针

【例7.14】

使用二级指针变量间接访问二维数组元素的地址和值。

7.3.5 指向指针的指针

例7.14

```
1 #include <stdio.h>
2 int main()
3 {
4   int a[4][4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
5   int *s[4]={a[0],a[1],a[2],a[3]}; //一维指针数组初始化
6   int i=2,j=3, **pp=s; //二级指针指向一维指针数组
7   printf("%x\t%x\t%x\n", &a[i][0], *(pp+i), pp[i]);
8   printf("%d\t%d\t%d\n", a[i][0], **(pp+i), pp[i][0]);
9   printf("%x\t%x\t%x\n", &a[i][j], *(pp+i)+j, pp[i]+j);
10  printf("%d\t%d\t%d\n", a[i][j], *(*(pp+i)+j), pp[i][j]);
11  return 0;
12 }
```

7.3.5 指向指针的指针

例7.14

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

程序运行屏幕

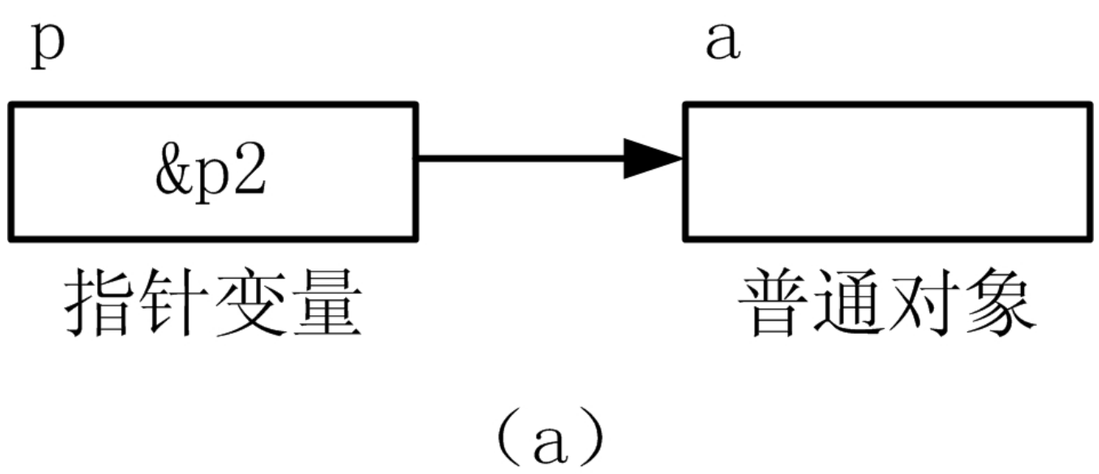
```
12ff60  12ff60  12ff60
9       9       9
12ff6c  12ff6c  12ff6c
12      12      12
```

```
][j], *(*(pp+i)+j), pp[i][j]);
```

7.3.5 指向指针的指针

- 通过指针变量间接访问对象，指针变量中存放的是目标对象的地址，称为间接寻址（简称间址）。

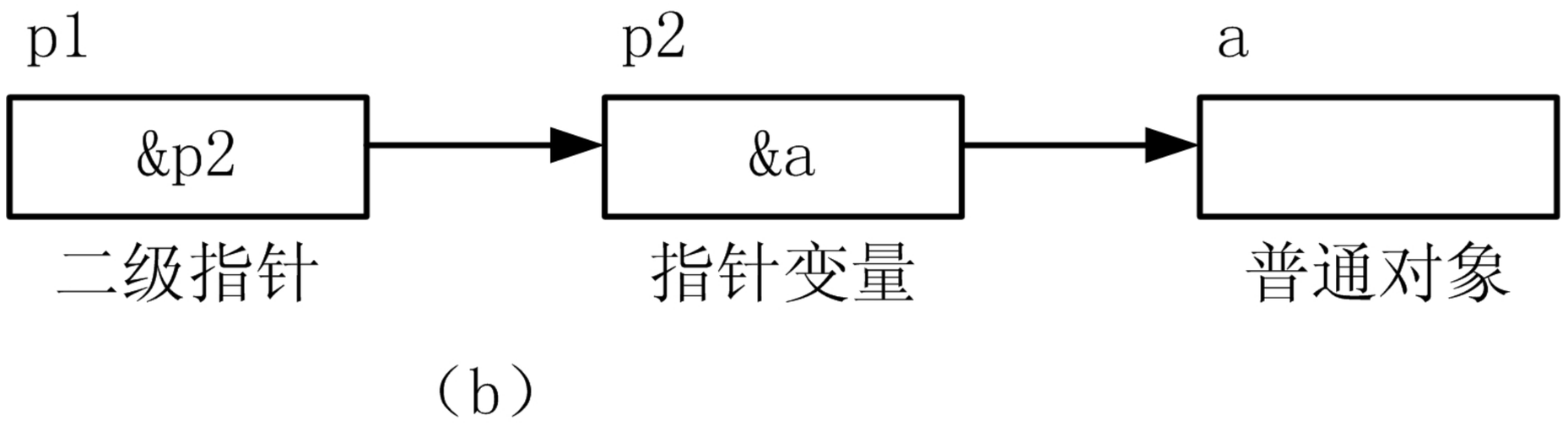
图7.17 多级指针的含义



7.3.5 指向指针的指针

- 指向指针的指针称为二级间址。

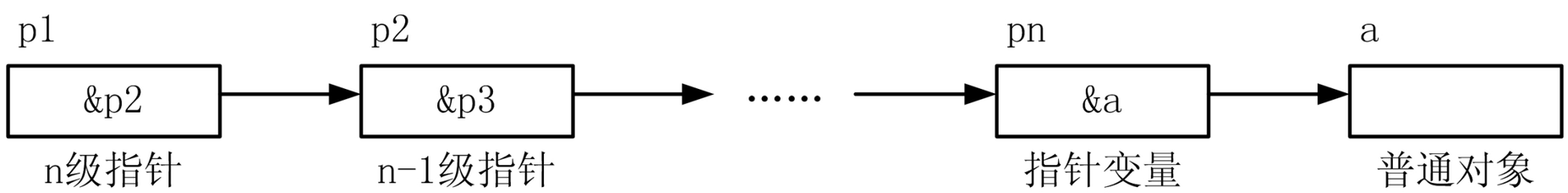
图7.17 多级指针的含义



7.3.5 指向指针的指针

- 多级指针形成了多级间址。

图7.17 多级指针的含义



（c）

7.3.5 指向指针的指针

- 级数越多，间接访问就愈难理解，编程就愈复杂，产生混乱和出错的机会也多，故实际编程中很少有超过二级间址的。


程序设计