如何实现基于内存地址的数据存取

张华

64174234@qq.com

内容

1.1 内存与地址

1.2 如何访问内存

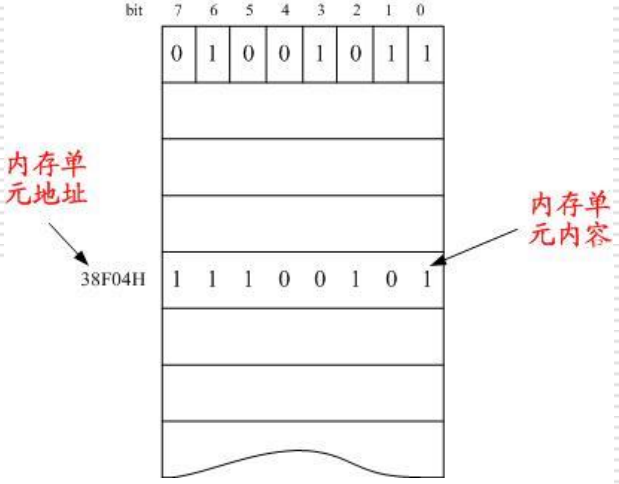
1.3 总结

1.1内存与地址

- 内存计算机内的存储部件，活动中的所有指令和数据都保存在内存内
- 速度快，但是掉电即失
- 可以随机访问
  - 只要指名要访问的内存单元的地址，就可以立即访问到该单元
  - 地址是一个无符号整数（通常用16进制数），其字长与主机相同
  - 内存中的每个字节都有唯一的一个地址

1.2如何访问内存

- Go 具有指针。 指针保存了变量的内存地址




```go
package main
import "fmt"
func main() {
    i, j := 42, 2701
    var p *int = &i //point to i, same as   p := &i
    fmt.Println(*p) // read i through the pointer
    *p = 21         // set i through the pointer
    fmt.Println(i)  // see the new value of i
    fmt.Println(p)  //see the address of i
    p = &j          // point to j
    *p = *p / 37    // divide j through the pointer
    fmt.Println(j)  // see the new value of j
    fmt.Println(p)  //see the address of j
    fmt.Println(&p) //see the address of p
}
```

1.3总结

- 类型 *T是指向类型T的指针
  - 零值是 nil
  - var p *int 定义指针变量p
- & 符号会生成一个指向其作用对象的指针，
  - i := 42
  - p=&i//取变量i的地址赋给指针变量p
- * 表示指针所指向的地址空间的值
  - fmt.Println(*p)//通过指针p读取i的值
  - *p=21//通过指针p设置i的值

Thank you very much

Any comments and suggestions are beyond welcome