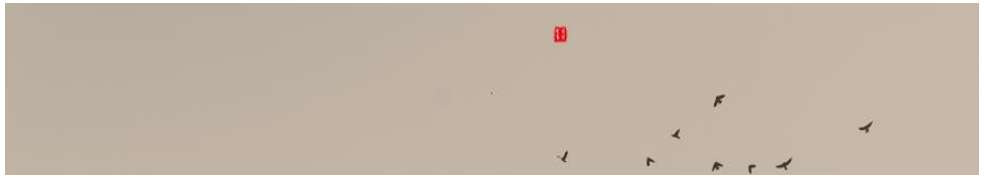
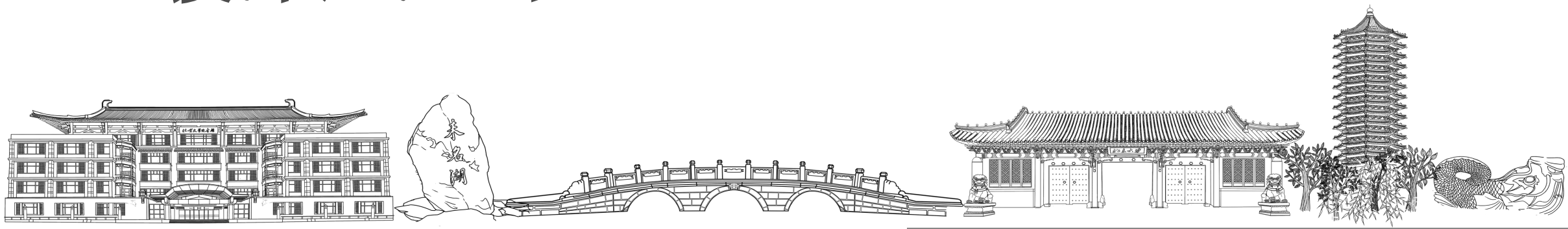
北京大学
PEKING UNIVERSITY



08

联合
内存分配
缓冲区溢出



联合

- 一个联合总的大小等于它最大字段的大小
- 某一时刻只能访问其中的一个成员变量

应用

- 减小分配空间的总量
- 转换

```
double uu2double(unsigned word0, unsigned word1)
{
    union {
        double d;
        unsigned u[2];
    } temp;
    temp.u[0] = word0;
    temp.u[1] = word1;
    return temp.d;
}
```

- 访问不同数据类型的位模式

```
double uu2double(unsigned word0, unsigned word1)
{
    union {
        double d;
        unsigned u[2];
    } temp;
    temp.u[0] = word0;
    temp.u[1] = word1;
    return temp.d;
}
```

- 将不同数据类型结合到一起时，注意字节顺序

内存分配

Stack：运行时栈（限制最高8MB，可手动改变），可存局部变量

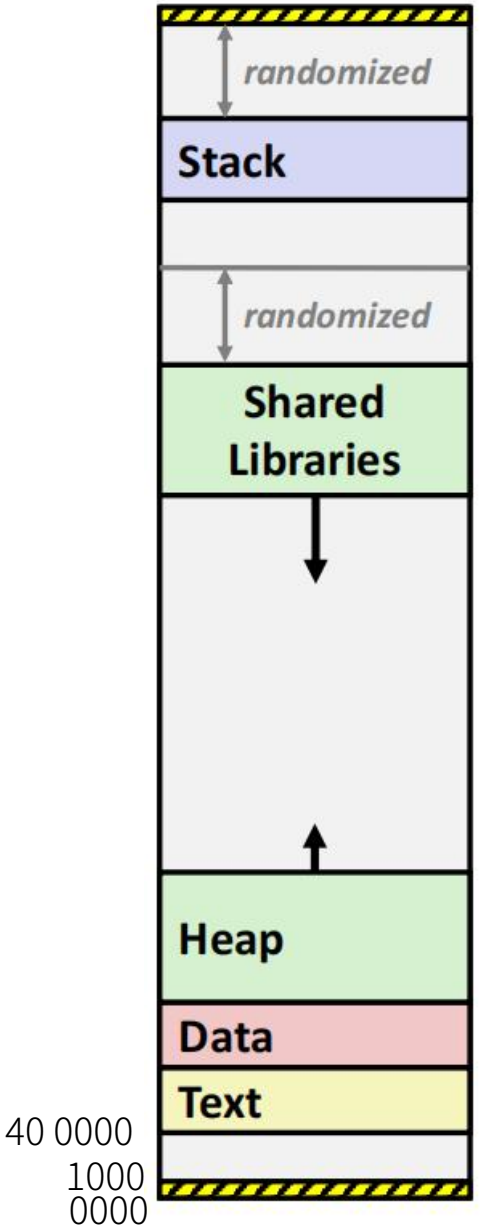
Heap：**动态内存分配，如malloc ()，calloc ()，new ()** 出的变量

分为两块，一般比较小的变量从下往上，较大的变量从上往下

可以提高运行效率

Data：存储静态变量，包括全局变量，静态变量，字符常量

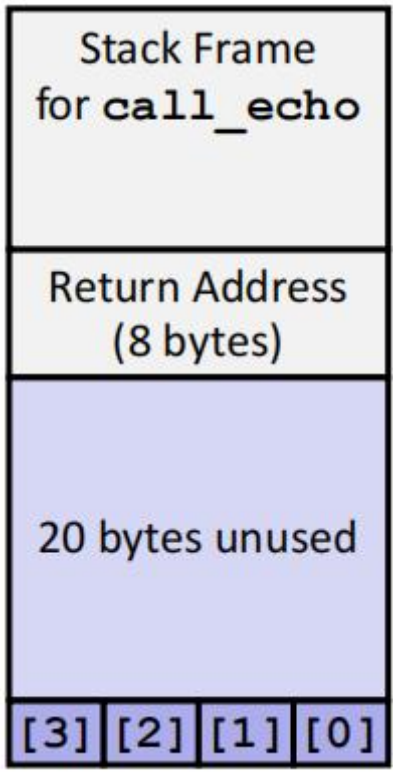
Text&Shared Libraries：存储机器代码，只读



缓冲区溢出 ►►►

北京大学 PEKING UNIVERSITY

- 向缓冲区写入超过其容量的数据，可能覆盖相邻内存



buf<--%rsp

```
1 void echo()
2 {
3   char buf[4];
4   gets(buf);
5   puts(buf);
6 }
```

```
1 echo:
2   subq  $24,%rsp
3   movq  %rsp,%rdi
4   call  gets
```

让程序执行它本不愿执行的函数

对抗缓冲区溢出攻击

北京大学 PEKING UNIVERSITY

- **栈随机化**

实现方式　程序开始时，在栈上分配一段随机大小的空间，如使用分配函数alloca在栈上分配指定字节数量的空间。

地址空间布局随机化（ASLR）

空操作雪橇（nop sled）---可以破解栈随机化的缓冲区溢出攻击方式

思考：对于地址随机范围为2^13，如果尝试一个有128字节nop sled的缓冲区溢出，要想穷尽所有的起始地址，需要尝试多少次？

- **栈破坏检测**
- **限制可执行代码区域**

对抗缓冲区溢出攻击

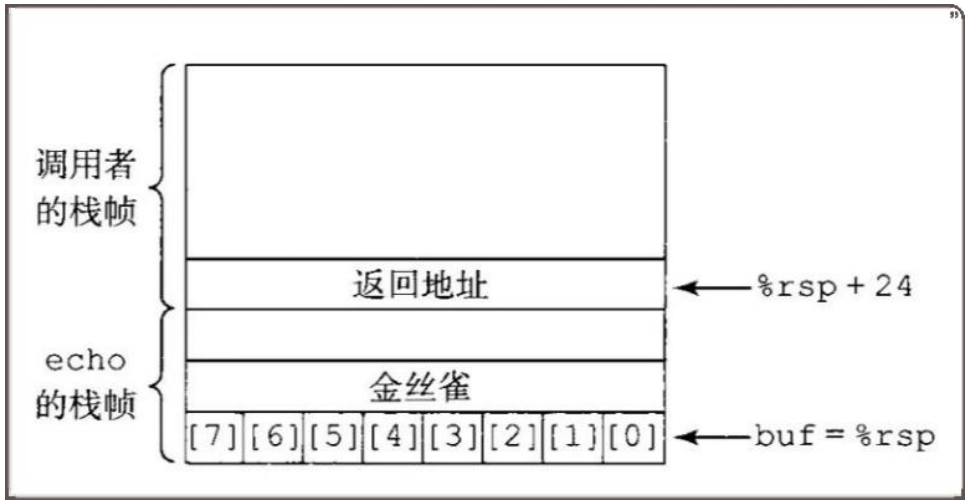
北京大学 PEKING UNIVERSITY

- 栈随机化

- **栈破坏检测**

“金丝雀值”---程序随机产生，攻击者无法得知。若发生改变，则程序终止。

（只读，攻击者无法修改）



- **限制可执行代码区域**

---消除攻击者向系统中插入可执行代码的能力


谢 谢
Thank You
北京大学
PEKING UNIVERSITY