Linux任督二脉之内存管理(一)

讲解时间：6月10日-14日晚9点

宋宝华 <21cnbao@gmail.com>

扫描二维码报名

麦当劳喜欢您来，喜欢您再来

扫描关注

Linux阅码场

硬件原理和分页管理

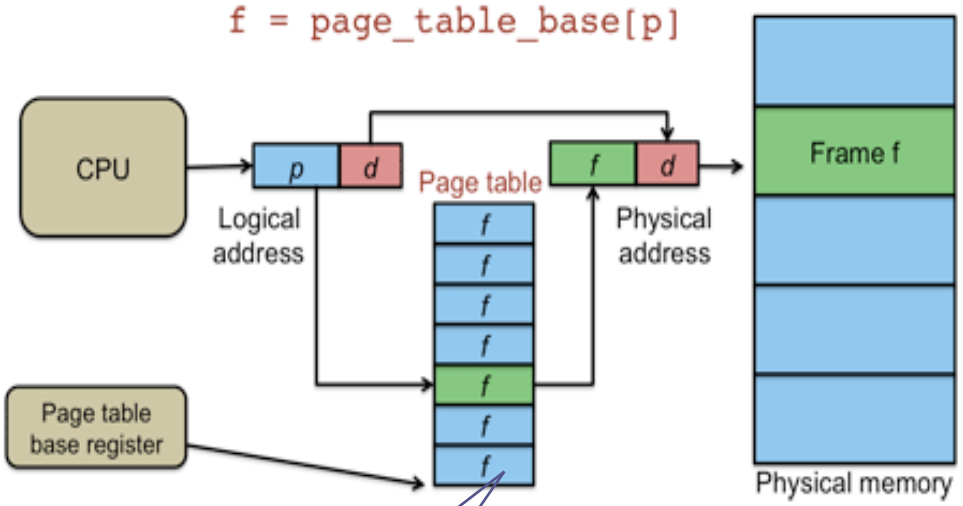
*CPU寻址内存，虚拟地址、物理地址
*MMU以及RWX权限、kernel和user模式权限
*内存的zone: DMA、Normal和HIGHMEM
*Linux内存管理Buddy算法
*连续内存分配器(CMA)

练习题

*尝试去更改一个const变量
*meltdown实例
*看/proc/buddyinfo

分页机制

1. 进程访问虚拟地址v=(p,d)
2. MMU以p作为索引检索页表
3. Page frame(f)，加上偏移d，得到物理地址



页表里面可以表明：RWX权限

页表里面可以表明：kernel/User+kernel权限

找不到物理地址
权限不对
都page fault!

尝试更改一个const变量

main.c

```
#include <stdio.h>

const int g =2;

extern void add_g(void);

main()
{//          g = 1;

            printf("%s %d\n", __func__, g);
            add_g();
            printf("%s %d\n", __func__, g);
}
```

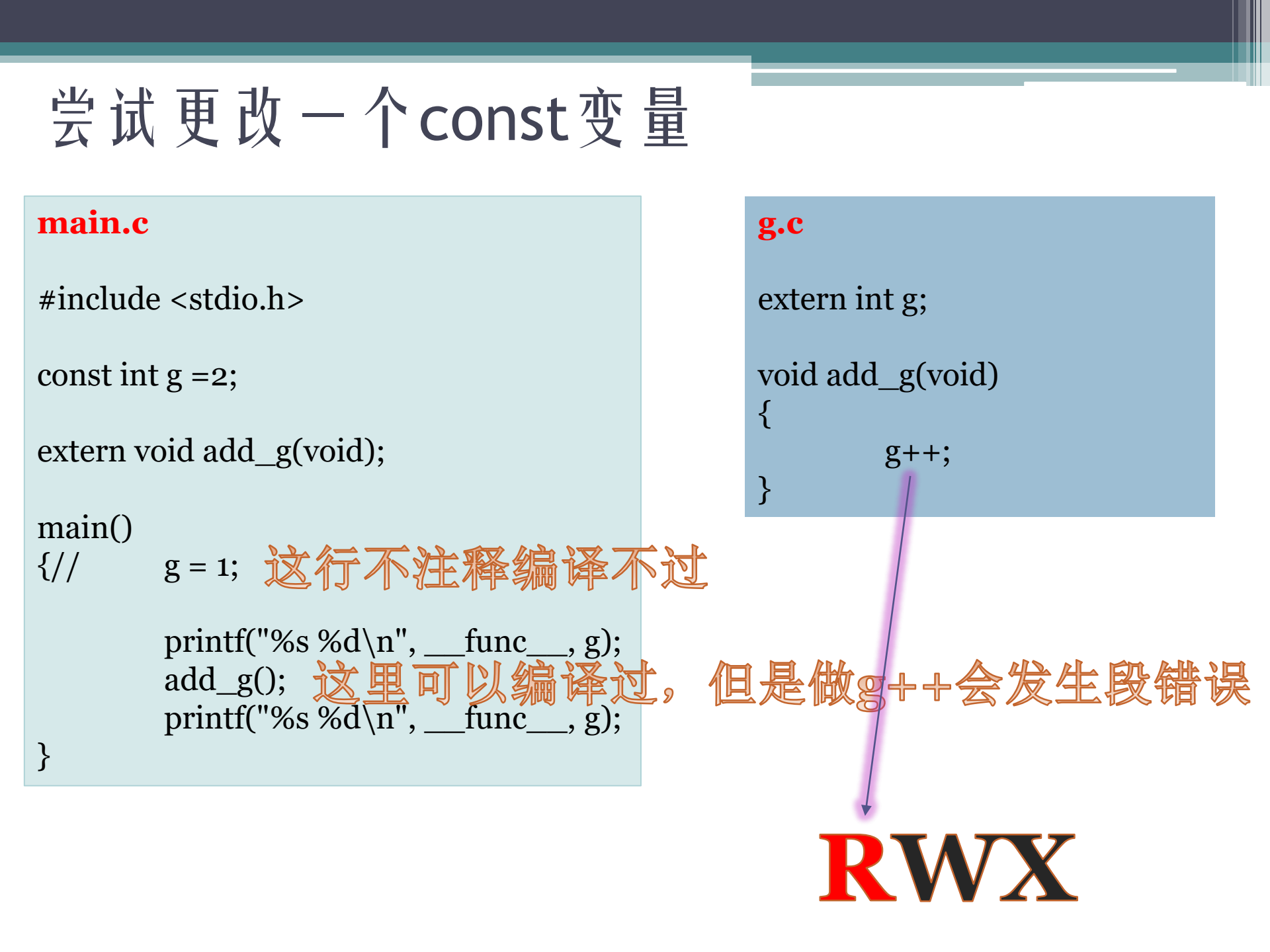
这行不注释编译不过

这里可以编译过，但是做g++会发生段错误

g.c

```
extern int g;

void add_g(void)
{
            g++;
}
```



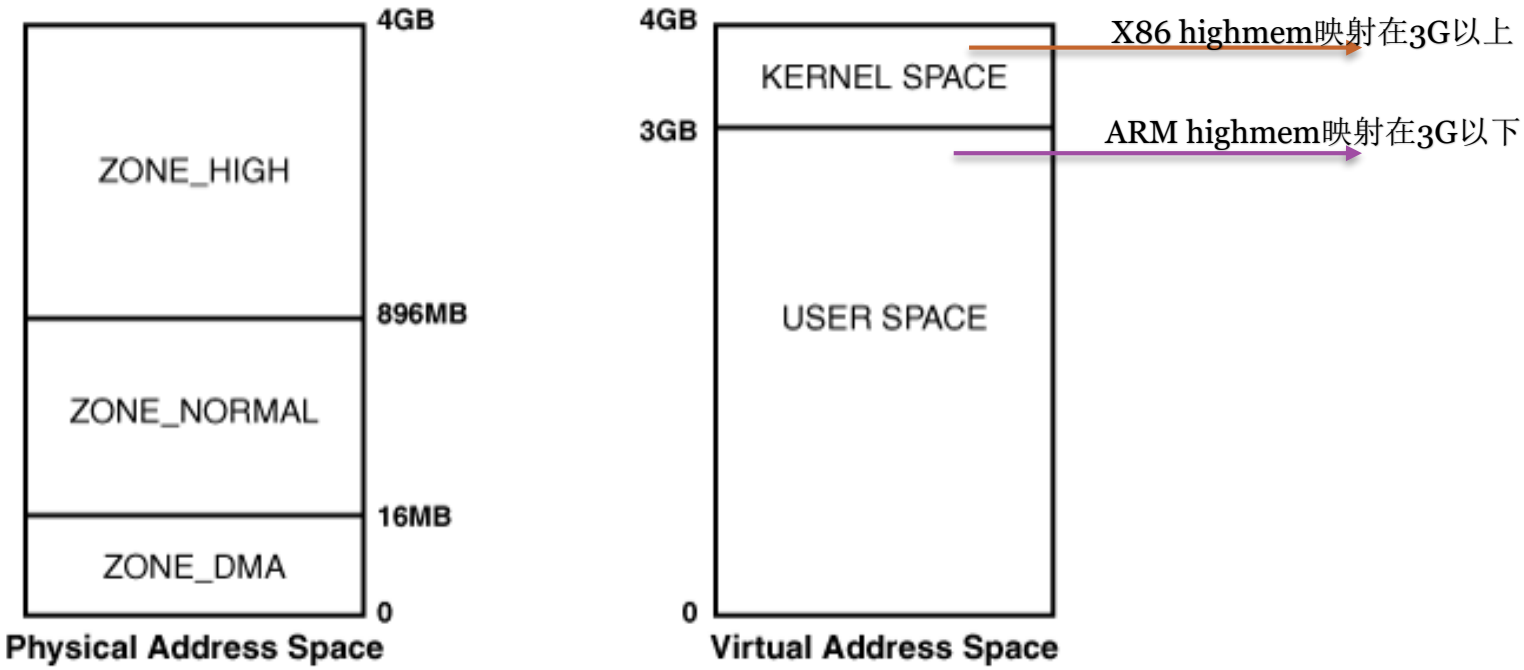
Meltdown漏洞

页表里面可以表明：kernel/User+kernel权限

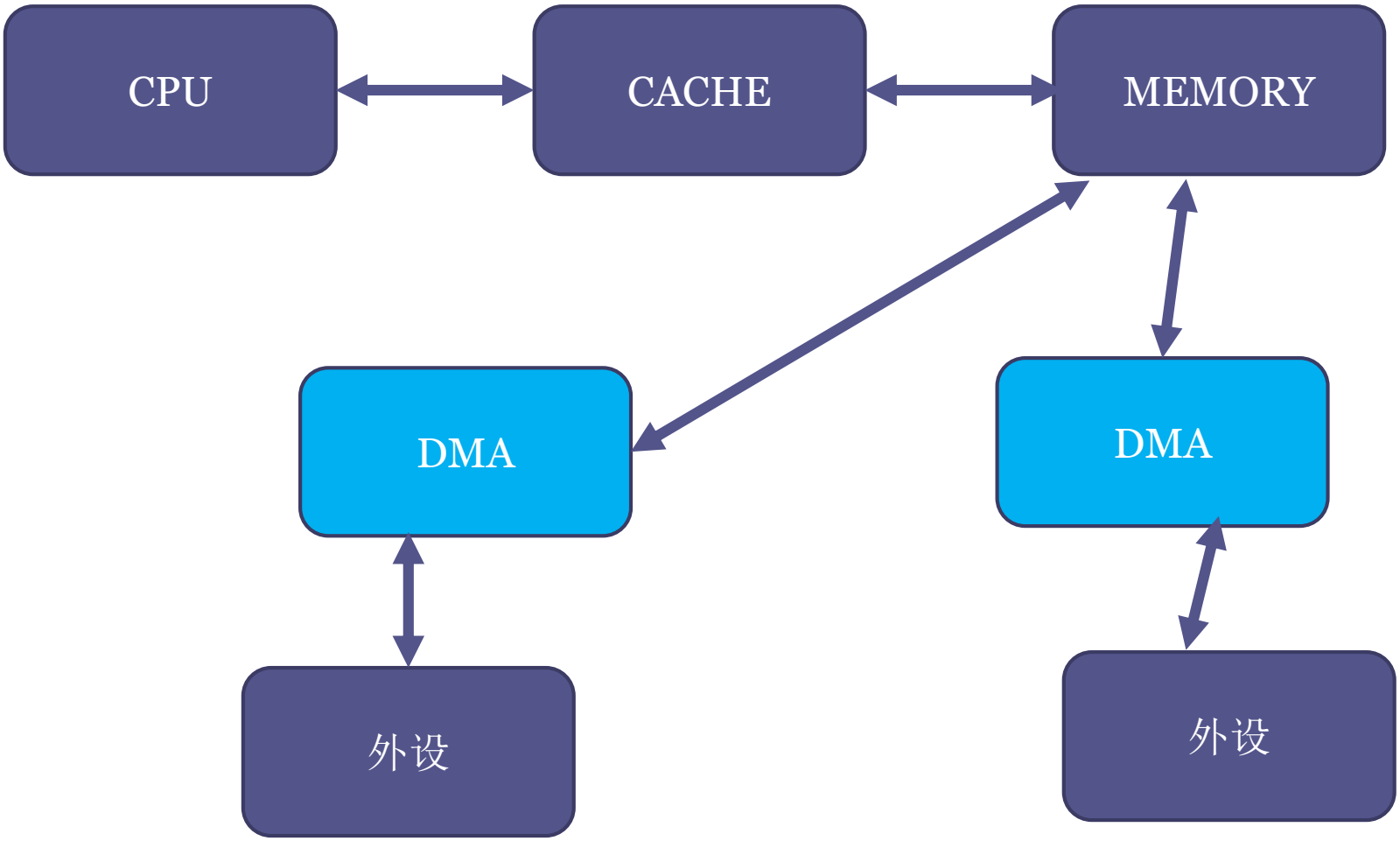
Meltdown则从用户空间偷取了内核空间数据

内存分ZONE的一般概念

DMA+ZONE往3G以上一一映射；
HIGHMEM内核一般不使用，如果要使用通过kmap映射；
DMA ZONE的原因是：有的硬件DMA引擎无法access所有内存。

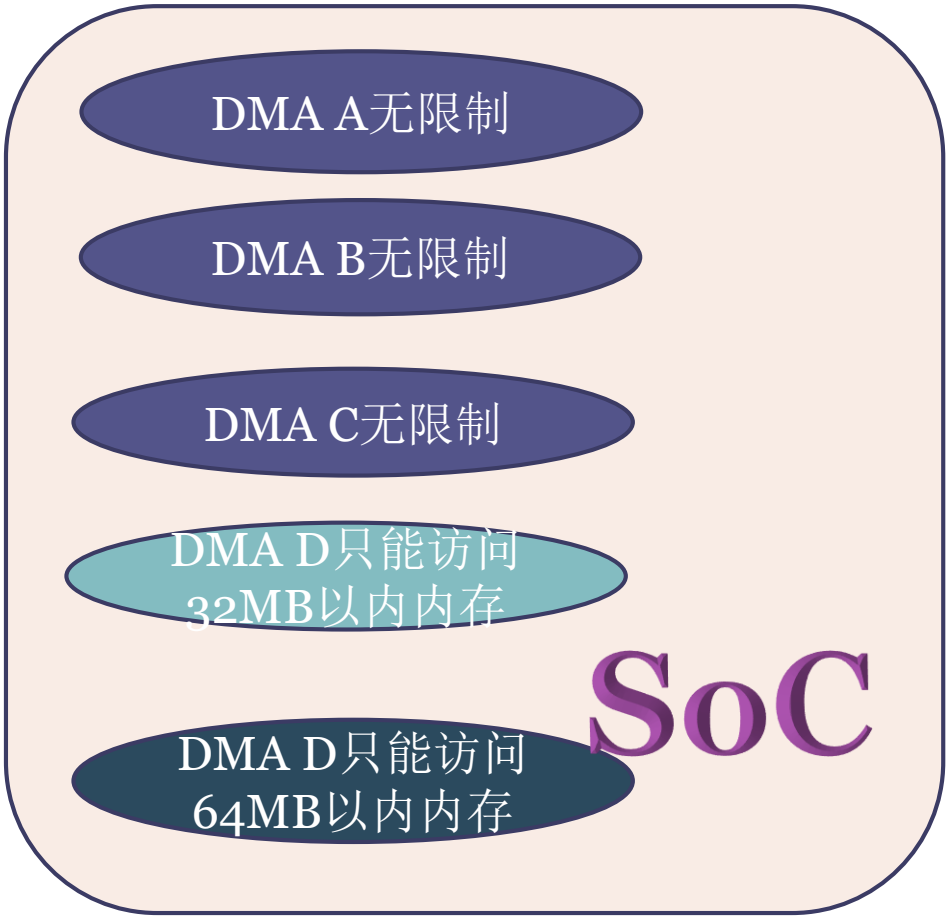


DMA：可以直接在内存和外设间进行数据搬移



DMA zone应该多大的例子

假设一款芯片，含多个DMA



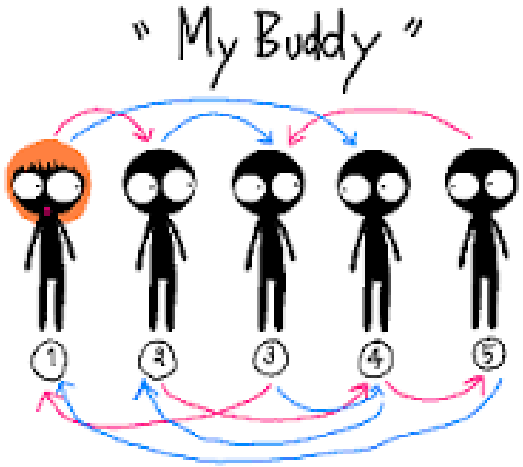
DMA ZONE设置多大？

---- 32MB !!

Buddy

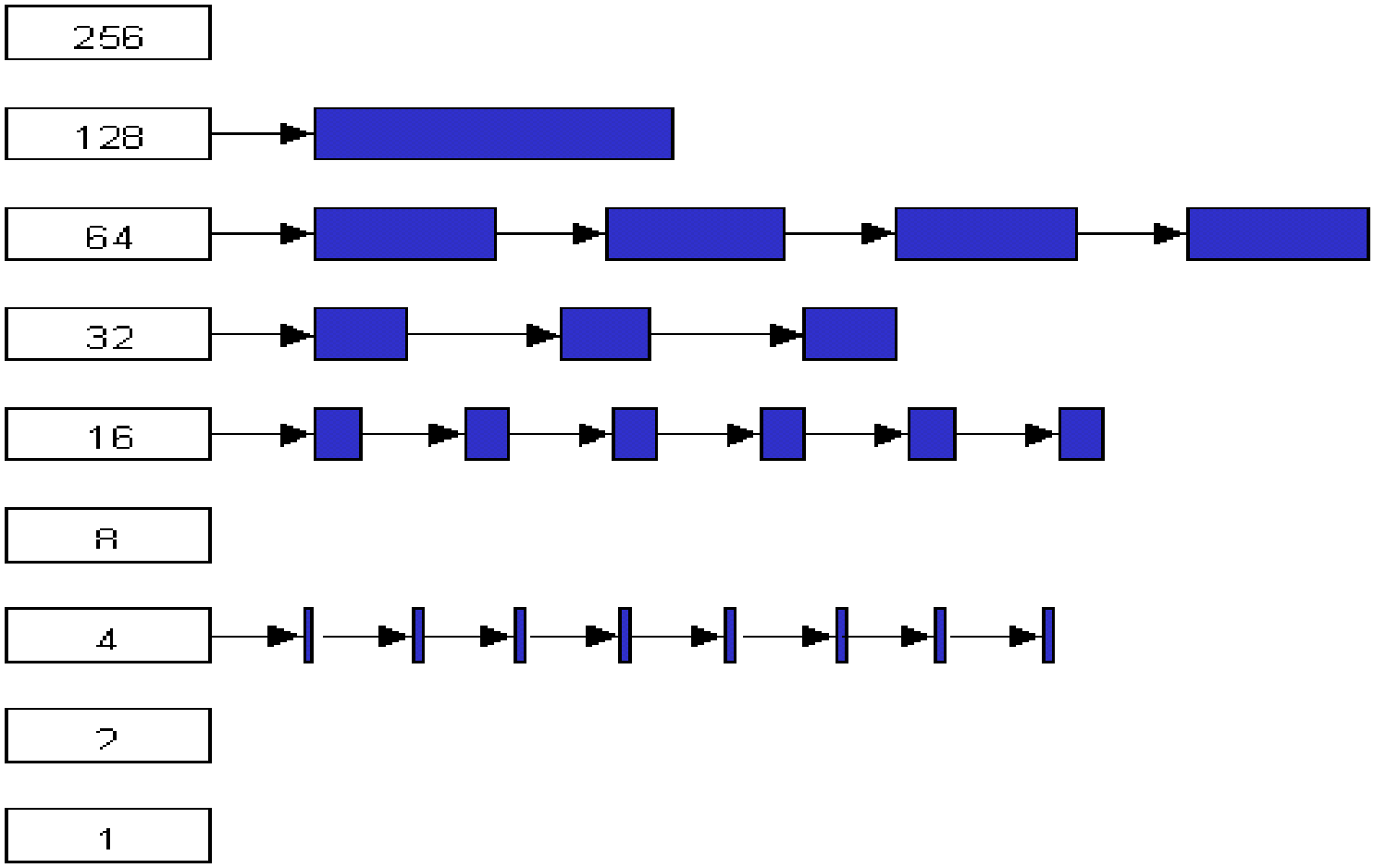
n. 密友，好友; 同伴，搭档; 互相帮助的朋友; [名]巴迪;
vi. 交朋友; 做朋友;

the buddy system was invented in 1963 by Harry Markowitz, who won the 1990 Nobel Memorial Prize in Economics



Linux Buddy分配算法

空闲页面按照2的n次方管理



Buddy info

DMA ZONE里面1页空闲的还有6个

```
baohua@baohua-VirtualBox:~$ cat /proc/buddyinfo
Node 0, zone      DMA      6      4      1      0      1      3      3      3      0      1      0
Node 0, zone   Normal    100     32     28     27     11     17      8      1      4      3     72
Node 0, zone  HighMem     46      9     16      5     12     10      4      6      1      3      6
```

NORMAL ZONE里面2页空闲的还有32

Buddy算法会导致内存碎片化

空闲内存很多，但是连续的空闲内存少！
谁需要连续的空闲内存？ DMA

一般方法: reserved内存
高级方法： CMA（连续内存分配器）

平时给app，movable页面用

平时不浪费！！

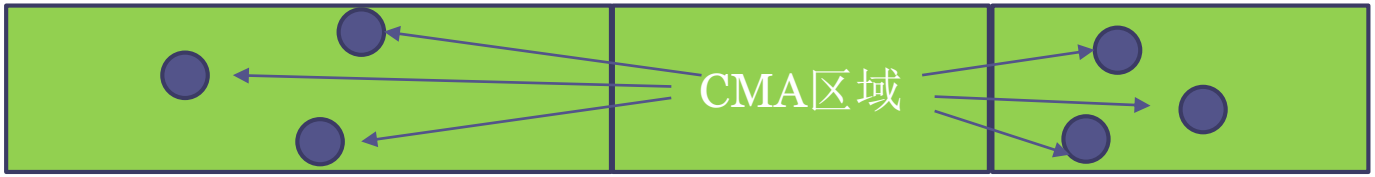
CMA的工作机制

平时给app，movable页面用

第一步

DMA要用了

第二步



漫山遍野散开去

课程练习源码

https://github.com/21cnbao/memory-courses

更早课程

- 《Linux总线、设备、驱动模型》录播：
http://edu.csdn.net/course/detail/5329

- 深入探究Linux的设备树
http://edu.csdn.net/course/detail/5627

- C语言大型软件设计的面向对象
https://edu.csdn.net/course/detail/6496

谢 谢！