

操作系统

Operating Systems

L23 段页结合的实际内存管理

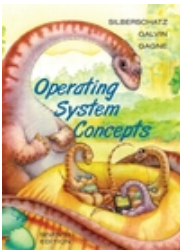
Segmentation & Paging

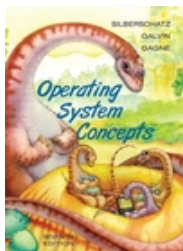
lizhijun_os@hit.edu.cn

综合楼411室

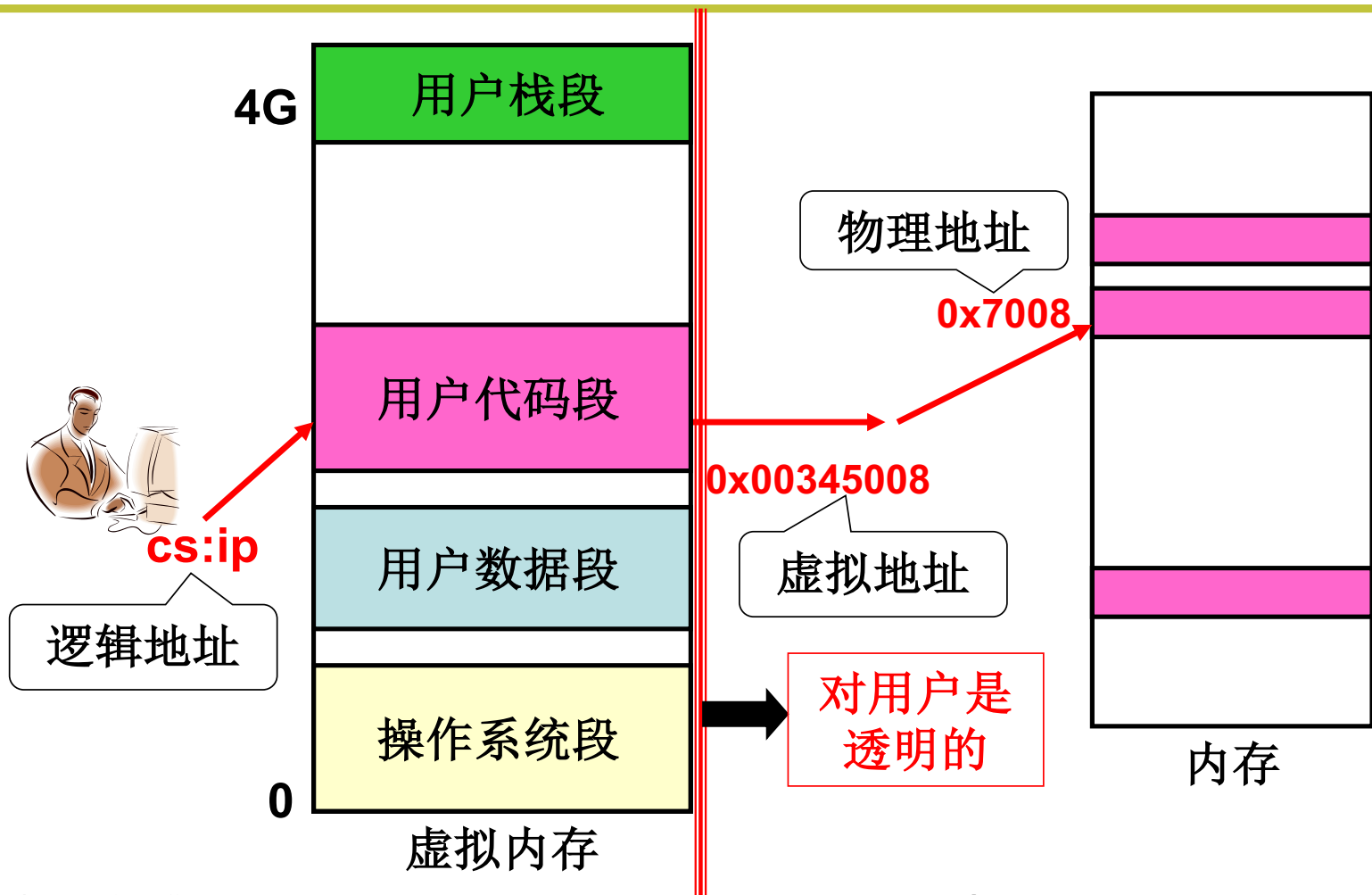
授课教师：李治军

**段、页结合：程序员希望用段，
物理内存希望用页，所以...**





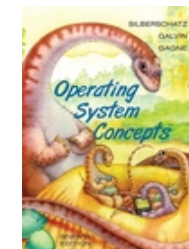
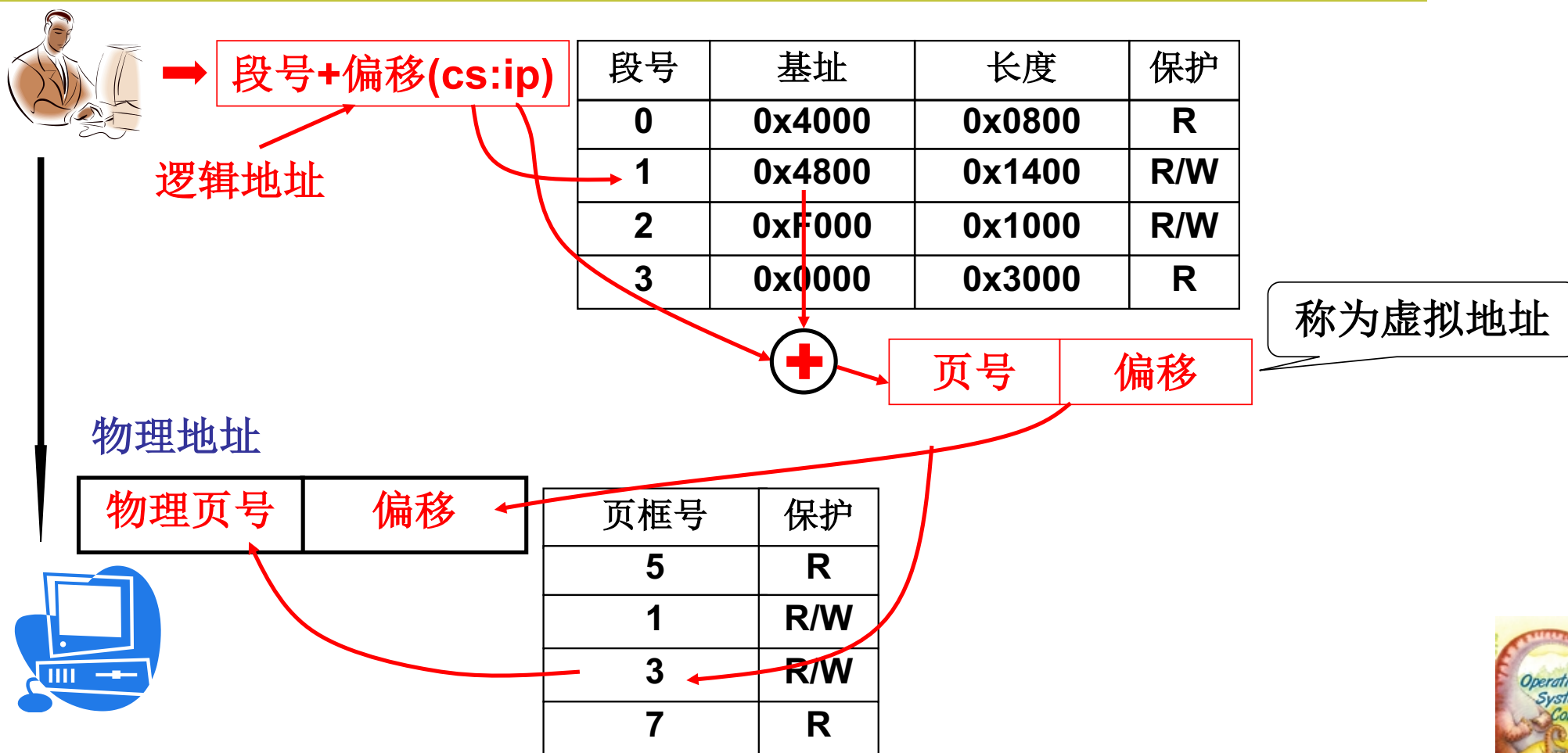
段、页同时存在：段面向用户/页面向硬件



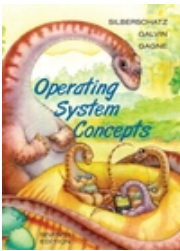
问题：为什么称为
虚拟内存？



段、页同时存在是的重定位(地址翻译)

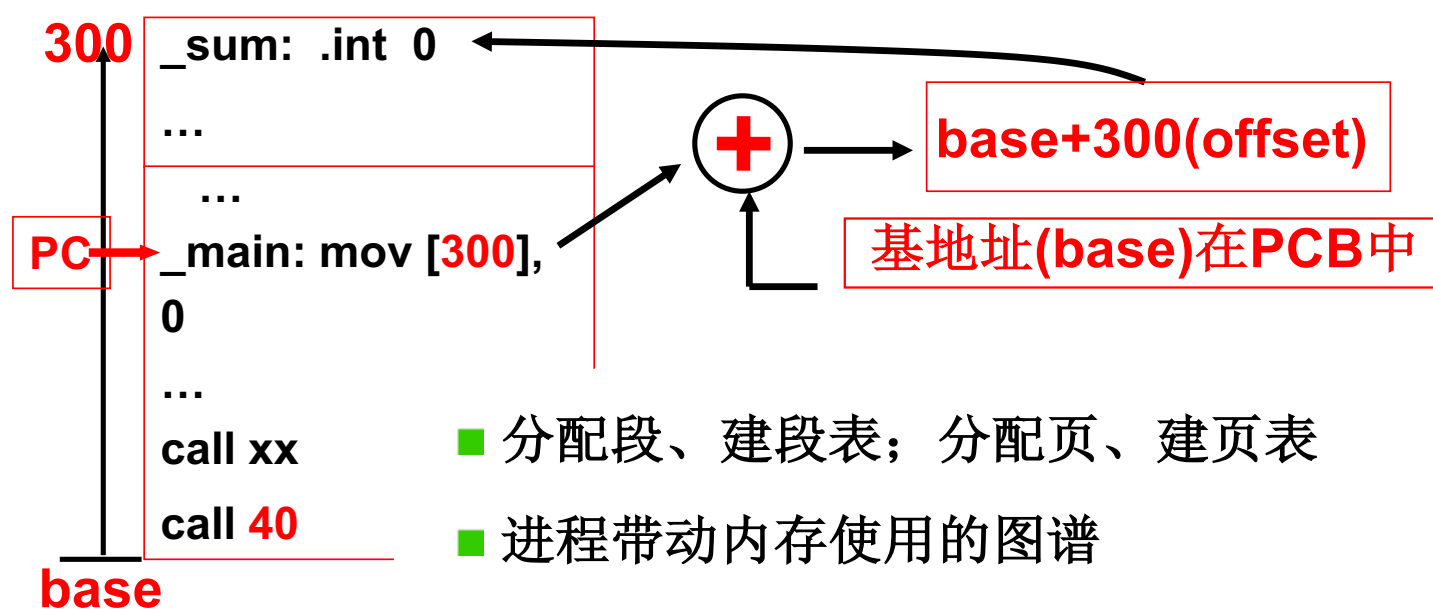


一个实际的段、页式内存管理

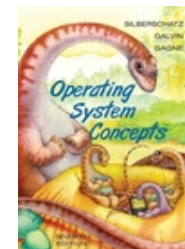
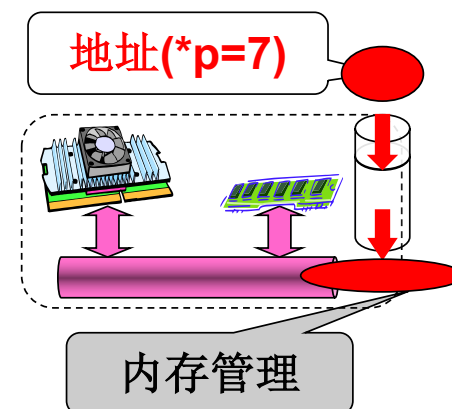


这个故事从哪里开始？

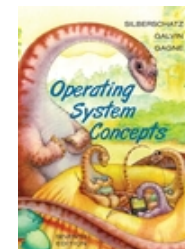
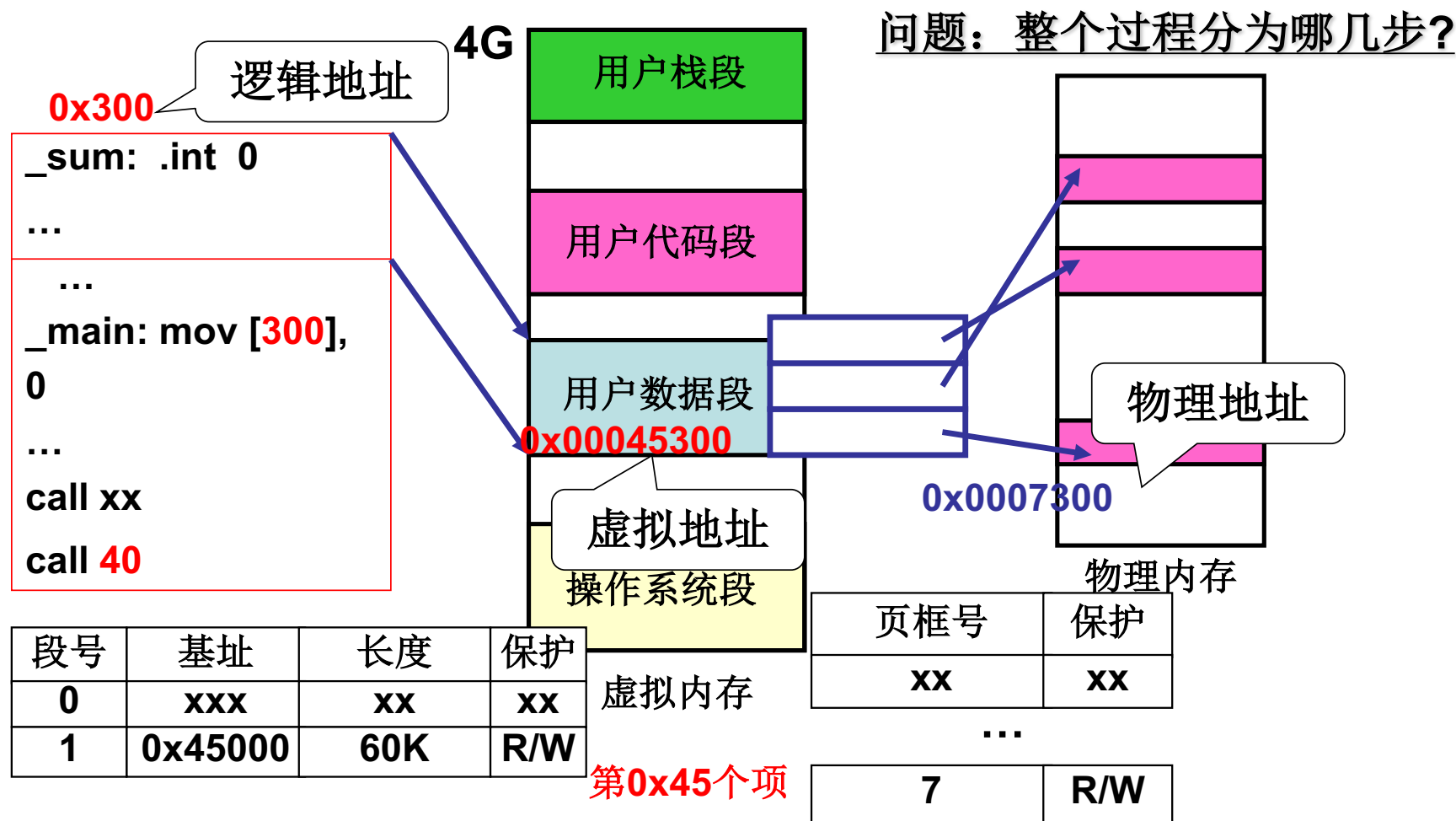
- 内存管理核心就是内存分配，所以从程序放入内存、使用内存开始...



- 分配段、建段表；分配页、建页表
- 进程带动内存使用的图谱
- 从进程fork中的内存分配开始..



段、页式内存下程序如何载入内存？



故事从fork()开始 分配虚存、建段表

■ fork()→sys_fork→copy_process的路都已经走过了

在linux/kernel/fork.c中

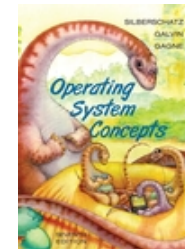
```
int copy_process(int nr, long ebp,...)
{
    ...
    copy_mem(nr, p); ...
}
```

的确是进程带动内存!

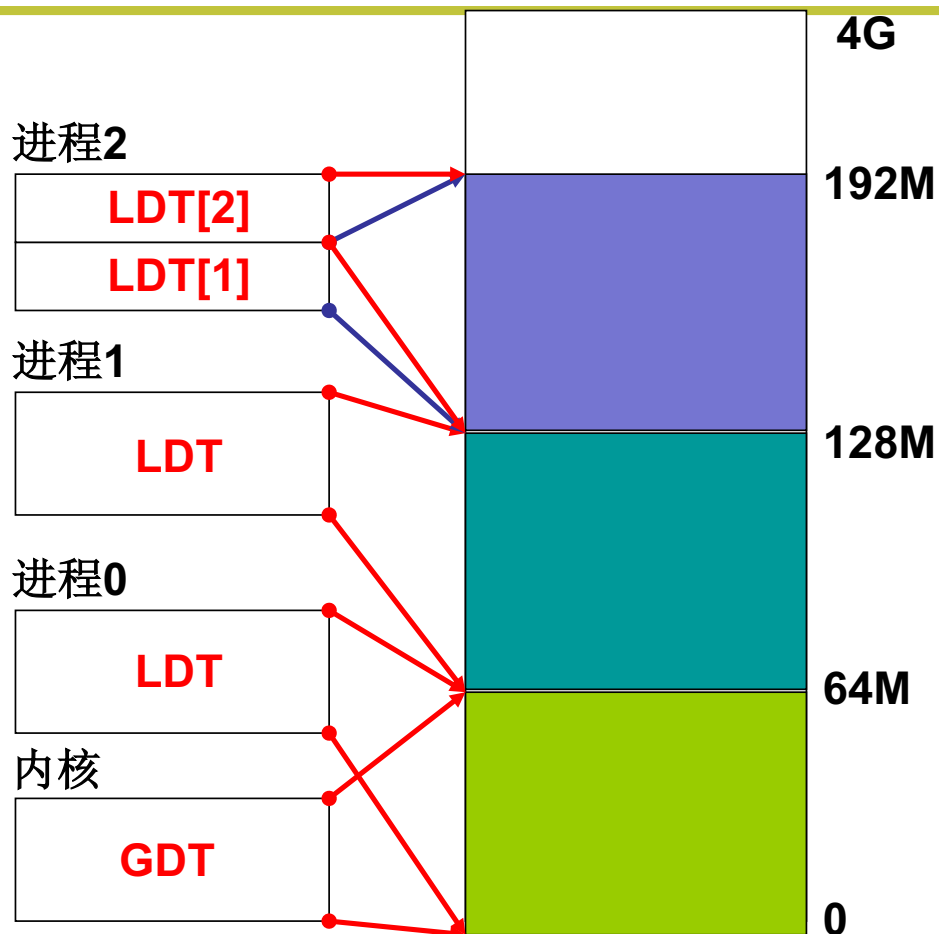
■ 现在开始分析当时那个神秘的copy_mem了

```
int copy_mem(int nr, task_struct *p)
{
    unsigned long new_data_base;
    new_data_base=nr*0x4000000; //64M*nr
    set_base(p->ldt[1],new_data_base);
    set_base(p->ldt[2],new_data_base);
}
```

p是什么?进程
切换跟着切换



进程0、进程1、进程2的虚拟地址



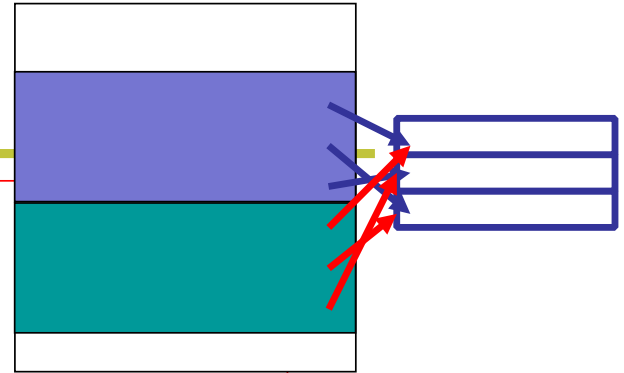
- 每个进程的代码段、数据段都是一个段
- 每个进程占**64M**虚拟地址空间，**互不重叠**

问题：这意味着什么样的简化？

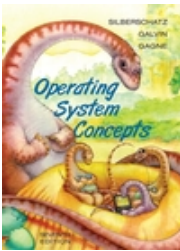


接下来应该是什么了？ 分配内存、建页表

```
int copy_mem(int nr, task_struct *p)
{
    unsigned long old_data_base;
    old_data_base=get_base(current->ldt[2]);
    copy_page_tables(old_data_base,new_data_base,data_limit);
}
```



```
int copy_page_tables(unsigned long from,unsigned long to, long size)
{
    from_dir = (unsigned long *) ((from>>20)&0xffc);
    to_dir = (unsigned long *) ((to>>20)&0xffc);
    size = (unsigned long) (size+0x3fffff)>>22;
    for(; size-->0; from_dir++, to_dir++){
        from_page_table=(0xffffffff000&*from_dir);
        to_page_table=get_free_page();
    }
}
```



这里的from_dir, to_dir是什么？

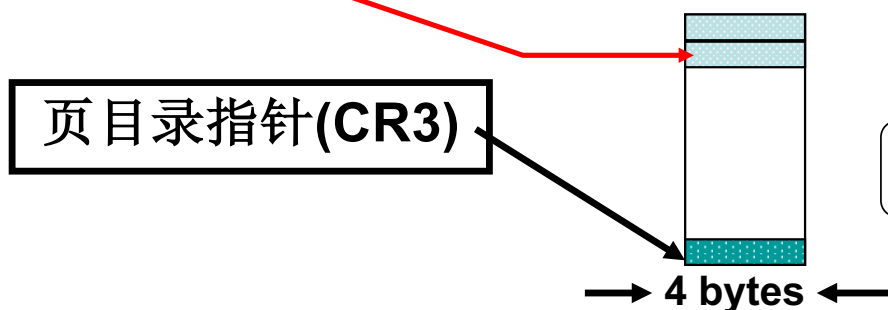
```
from_dir = (unsigned long *) ((from >> 20) & 0xffc);  
to_dir = (unsigned long *) ((to >> 20) & 0xffc);  
size = (unsigned long) (size + 0x3fffff) >> 22;
```

■ from是？ 32位虚拟地址，这个地址的格式是否还记得？



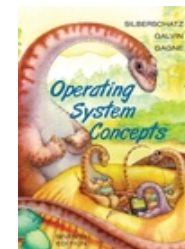
from >> 22 得到目录项编号

(from >> 22) * 4 每项4字节



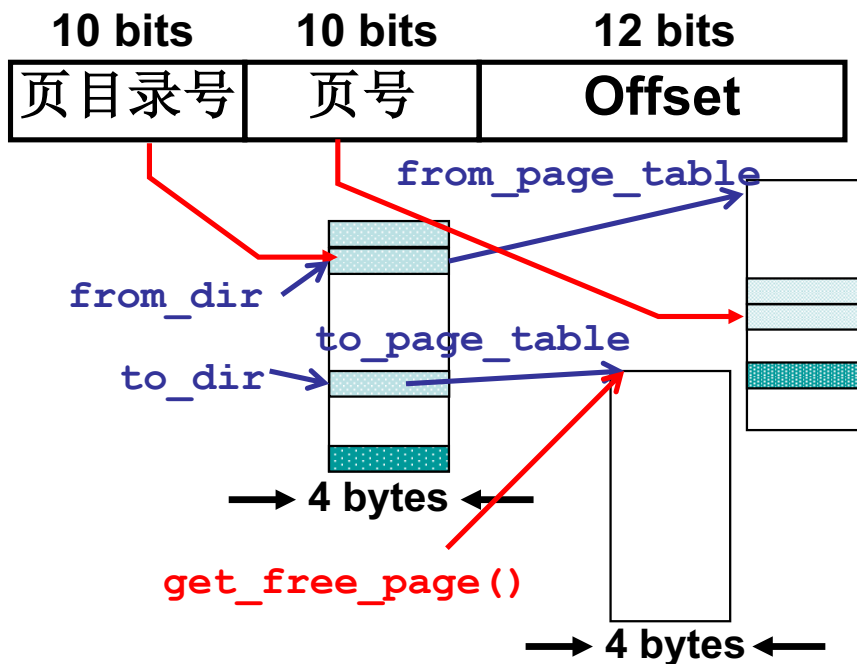
这要干什么？

```
for (; size-- > 0; from_dir++, to_dir++) {  
    from_page_table = (0xffffffff & *from_dir);
```

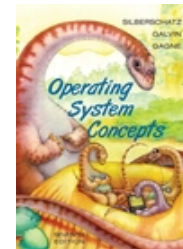
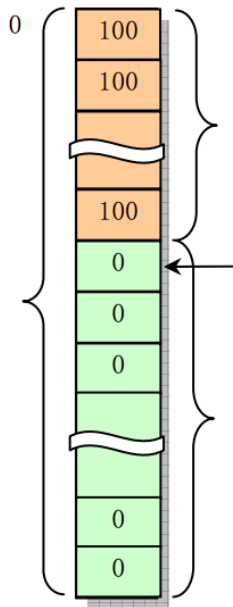


from_page_table与to_page_table?

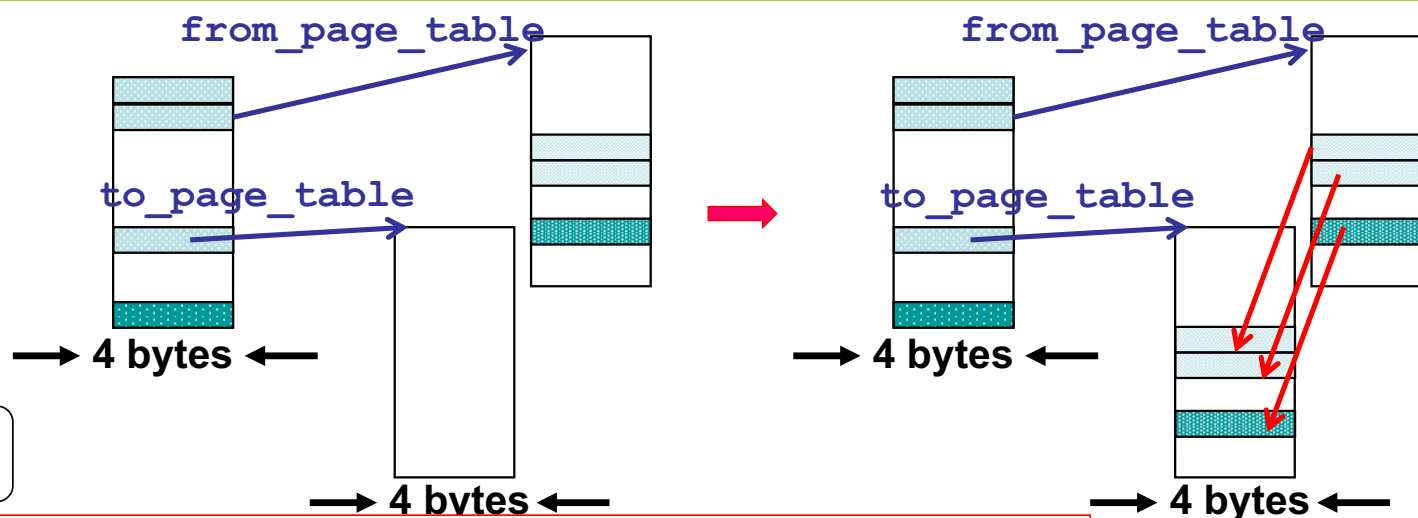
```
for(; size-->0; from_dir++, to_dir++){
    to_page_table=get_free_page();
    *to_dir=((unsigned long)to_page_table)|7;
```



```
unsigned long
get_free_page(voi
{ register unsign
long _res asm("ax
_asm_("std; repne
scasb\n\t"
"movl %%edx,%%eax
"D" (mem_map+PAGIG
GES-1));
return _res; }
```

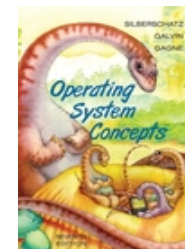


接下来干什么应该猜也猜的到...

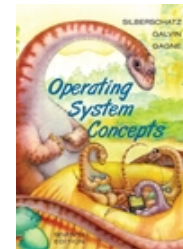
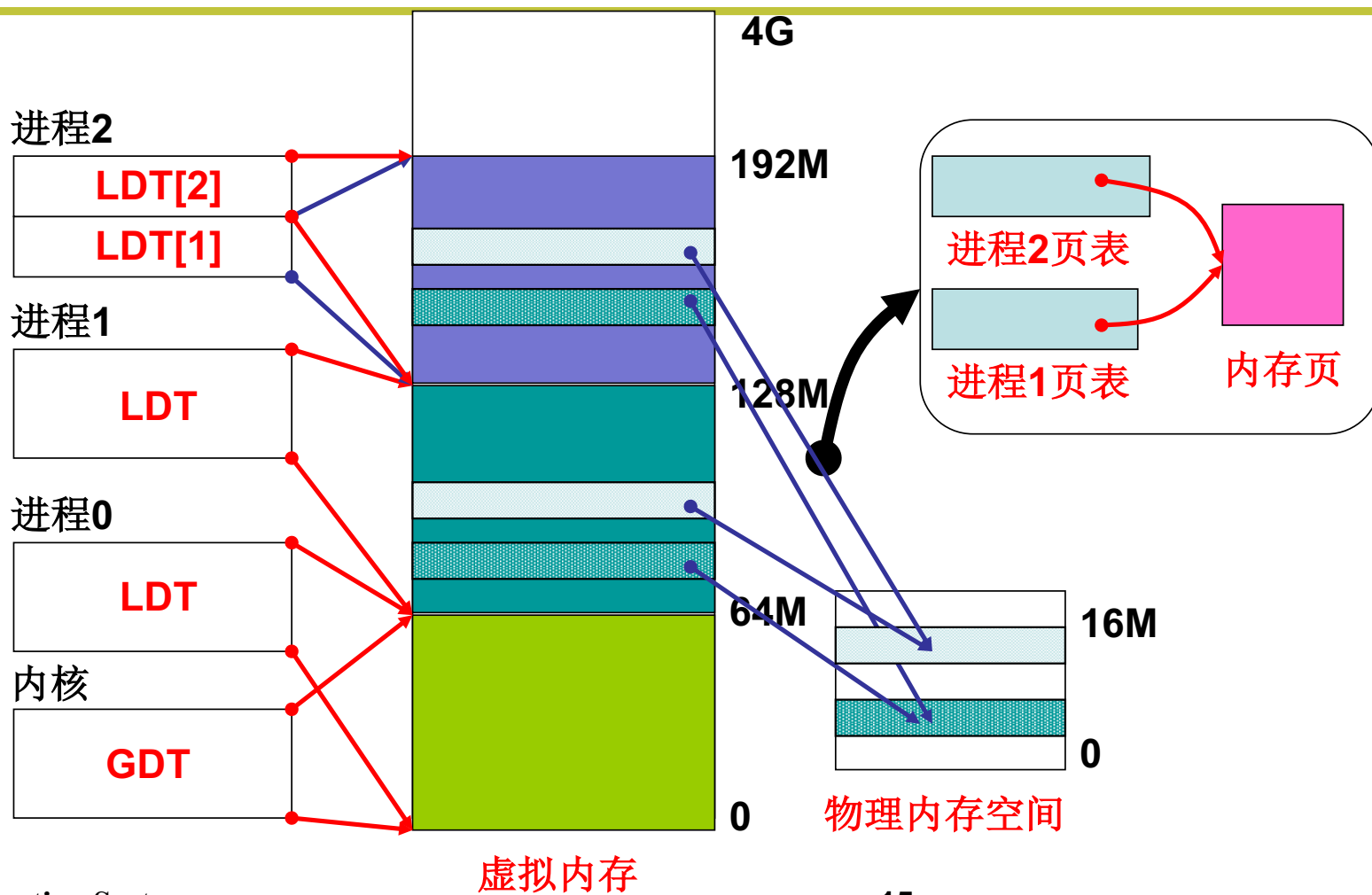


nr=1024!

```
for(;nr-->0;from_page_table++,to_page_table++){
    this_page = *from_page_table;
    this_page&=~2;//只读
    *to_page_table=this_page;
    *from_page_table=this_page;
    this_page -= LOW_MEM; this_page >>= 12;
    mem_map[this_page]++; }
```



程序、虚拟内存+物理内存的样子



*p=7? 父进程*p=7、子进程*p=8? 读写内存 *p=7

