

哈工大操作系统-L24请求调页内存换入

哈工大操作系统-L24请求调页内存换入

- 1.换入换出实现大内存
- 2.换入
- 3.换入的实际实现

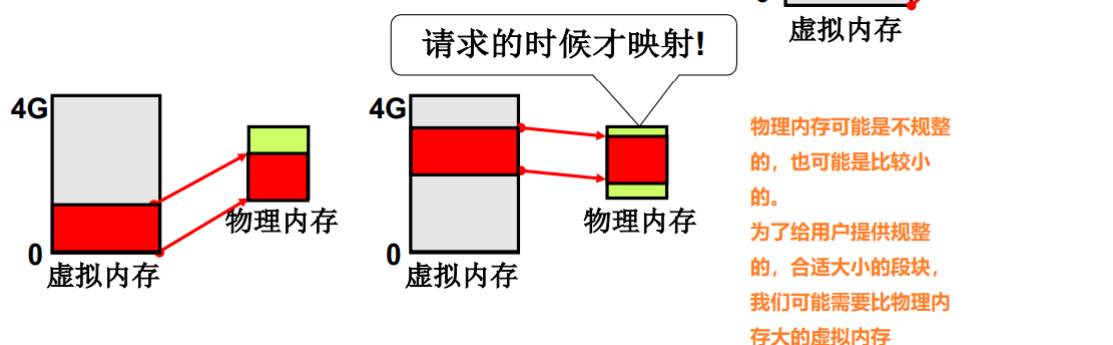
没有换入换出，虚拟内存也是做不了的。

1.换入换出实现大内存

用换入、换出实现“大内存”

■ 左边4G，右边1G怎么办？

- 访问 $p(=0G - 1G)$ 时，将这部分映射到物理内存
- 再访问 $p(=3G-4G)$ 时，再映射这一部分

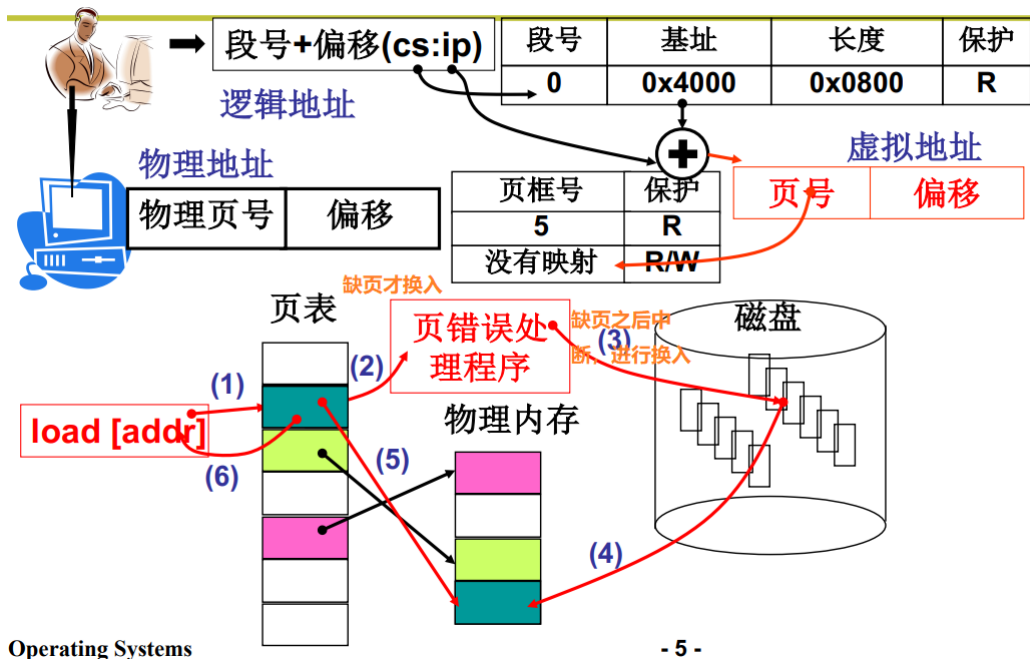


- 换入换出，实现大的虚拟内存，以为用户提供规整的、大小合适的段内存
 - 而实际的物理内存可能是没那么规整的，大小没有这么大的

2.换入

- 请求时才换入，建立映射
- 缺页使用中断把在磁盘中的页调入
- 缺页中断的话，可能会使PC不加1，然后中断回来继续执行原理的地址访问指令，因为页面已经调入，此时不会再发生中断了

请求调页



3.换入的实际实现

- 申请空闲页
- 把页从磁盘中读进来
- 建立映射

一个实际系统的请求调页

■ 这个故事从哪里开始？

■ 请求调页，当然从缺页中断开始

中断号	名称	说明
12	Segment not Present	描述符所指的段不存在
14	Page fault	页不在内存

```
void trap_init(void)
{ set_trap_gate(14, &page_fault); }

#define set_trap_gate(n, addr) \
    _set_gate(&idt[n], 15, 0, addr);
```

处理中断page fault

```
//在linux/mm/page.s中
.globl _page_fault
    xchgl %eax, (%esp)
    pushl %ecx
    pushl %edx
    push %ds
    push %es
    push %fs
    movl $0x10, %edx
    mov %dx, %ds
    mov %dx, %es
    mov %dx, %fs
    movl %cr2, %edx

    pushl %edx
    pushl %eax
    testl $1, %eax
    jne 1f
    call _do_no_page
    jmp 2f
1: call _do_wp_page //保护
2: add $8, %esp
    pop %fs
    pop %es
    pop %ds
    pop %edx
    pop %ecx
    pop %eax
    iret
    - 8 -
```

错误码被压到了栈中

压入参数

测试标志P

页错误线性地址

把发生错误的虚拟地址给edx

erating Systems

do_no_page

```
//在linux/mm/memory.c中
void do_no_page(unsigned long error_code,
                unsigned long address)
{
    address&=0xfffff000; //页面地址 得到虚拟页号
    tmp=address-current->start_code; //页面对应的偏移
    if(!current->executable||tmp>=current->end_data){
        get_empty_page(address); return; }
    page=get_free_page();
    bread_page(page, current->executable->i_dev, nr); //磁盘读写
    put_page(page, address);
}

void get_empty_page(unsigned long address)
{
    unsigned long tmp = get_free_page();
    put_page(tmp, address);
}
```

不是代码和数据!

读文件系统...

put_page

```
//在linux/mm/memory.c中
unsigned long put_page(unsigned long page, //物理地址
                        unsigned long address)
{ unsigned long tmp, *page_table;
  page_table=(unsigned long *)((address>>20)&ffc);
  if ((*page_table)&1) 找到页目录项
    page_table=(unsigned long*)(0xffffffff000&*page_table);
  else{
    tmp=get_free_page();
    *page_table=tmp|7;
    page_table=(unsigned long*)tmp;}
  page_table[(address>>12)&0x3ff] = page|7; 找到页表项, 并把传进
  return page; 来的物理页地址给页表
} 项
```

页目录项