实验五：实现系统调用

(10分)

实验目的：

1、学习掌握PC系统的软中断指令

2、掌握操作系统内核对用户提供服务的系统调用程序设计方法

3、掌握C语言的库设计方法

4、掌握用户程序请求系统服务的方法

实验要求：

1、了解PC系统的软中断指令的原理

2、掌握x86汇编语言软中断的响应处理编程方法

3、扩展实验四的的内核程序，增加输入输出服务的系统调用。

4、C语言的库设计，实现putch()、getch()、printf()、scanf()等基本输入输出库过程。

5、编写实验报告，描述实验工作的过程和必要的细节，如截屏或录屏，以证实实验工作的真实性

实验内容：

1. 修改实验4的内核代码，先编写save()和restart()两个汇编过程，分别用于中断处理的现场保护和现场恢复，内核定义一个保护现场的数据结构，以后，处理程序的开头都调用save()保存中断现场，处理完后都用restart()恢复中断现场。

保护现场的数据结构:

typedef strut {

int ax;

int bx;

int cx;

int dx;

int cs;

int ds;

int es;

int ss;

int sp;

int bp;

int di;

int si;

int ip;

int flag;

} cpuRegisters

;参考程序

;Minix的save和restart

; save

;=====================================================

save: ; save the machine state in the proc table.

push ds ; stack: psw/cs/pc/ret addr/ds

push cs ; prepare to restore ds

pop ds ; ds has now been set to cs

mov ds,ker\_ds ; word 4 in kernel text space contains ds value

pop ds\_save ; stack: psw/cs/pc/ret addr

pop ret\_save ; stack: psw/cs/pc

mov bx\_save,bx ; save bx for later ; we need a free register

mov bx,dgroup:proc\_ptr ; start save set up; make bx point to save area

add bx,OFF ; bx points to place to store cs

pop PC-OFF[bx] ; store pc in proc table

pop csreg-OFF[bx] ; store cs in proc table

pop PSW-OFF[bx] ; store psw

mov ssreg-OFF[bx],ss ; store ss

mov spreg-OFF[bx],sp ; sp as it was prior to interrupt

mov sp,bx ; now use sp to point into proc table/task save

mov bx,ds ; about to set ss

mov ss,bx ; set ss

push ds\_save ; start saving all the registers, sp first

push es ; save es between sp and bp

mov es,bx ; es now references kernel memory too

push bp ; save bp

push di ; save di

push si ; save si

push dx ; save dx

push cx ; save cx

push bx\_save ; save original bx

push ax ; all registers now saved

mov sp,offset dgroup:k\_stack ; temporary stack for interrupts

add sp,K\_STACK\_BYTES ; set sp to top of temporary stack

mov splimit,offset dgroup:k\_stack ; limit for temporary stack

add splimit,8 ; splimit checks for stack overflow

mov ax,ret\_save ; ax = address to return to

jmp ax ; return to caller; Note: sp points to saved ax

;==================================================

; restart

;==================================================

restart: ; This routine sets up and runs a proc or task.

cmp dgroup:cur\_proc,IDLE; restart user; if cur\_proc = IDLE, go idle

je \_idle ; no user is runnable, jump to idle routine

cli ; disable interrupts

mov sp,dgroup:proc\_ptr ; return to user, fetch regs from proc table

pop ax ; start restoring registers

pop bx ; restore bx

pop cx ; restore cx

pop dx ; restore dx

pop si ; restore si

pop di ; restore di

mov lds\_low,bx ; lds\_low contains bx

mov bx,sp ; bx points to saved bp register

mov bp,SPLIM-ROFF[bx] ; splimit = p\_splimit

mov splimit,bp ; ditto

mov bp,dsreg-ROFF[bx] ; bp = ds

mov lds\_low+2,bp ; lds\_low+2 contains ds

pop bp ; restore bp

pop es ; restore es

mov sp,spreg-ROFF[bx] ; restore sp

mov ss,ssreg-ROFF[bx] ; restore ss using the value of ds

push PSW-ROFF[bx] ; push psw (flags)

push csreg-ROFF[bx] ; push cs

push PC-ROFF[bx] ; push pc

lds bx,DWORD PTR lds\_low ; restore ds and bx in one fell swoop

iret ; return to user or task

1. 内核增加int 20h、int 21h和int 22h软中断的处理程序，其中，int 20h用于用户程序结束是返回内核准备接受命令的状态；int 21h用于系统调用，并实现3-5个简单系统调用功能；int22h功能未定，先实现为屏幕某处显示INT22H。

Int20h中断处理程序(用户程序结束是返回内核):

call save()

return\_kernel()

jmp restart ; 为了结构对称，不会执行此指令。

Int21h中断处理程序(系统调用):

call save()

call sys\_call()

jmp restart

Int22h中断处理程序(屏幕某处显示INT22H):

call save()

call print\_int22h()

jmp restart

1. 保留无敌风火轮显示，取消触碰键盘显示OUCH!这样功能。
2. 进行C语言的库设计，实现putch()、getch()、gets()、puts()、printf()、scanf()等基本输入输出库过程，汇编产生libs.obj。

例子：

public \_printf

\_printf proc

mov ax,0500h ;系统调用5号功能，显示输出一个字符串

push ax

… ； 其他参数进栈

int 21h ；产生中断，

\_printf end

public \_getch

\_getch proc

mov ax,0600h ;系统调用6号功能，从键盘输入一个字符

push ax

… ； 其他参数进栈

int 21h ；产生中断，

\_getch end

public \_gets

\_gets proc

mov ax,0700h ;系统调用7号功能，从键盘输入一个字符串

push ax

… ； 其他参数进栈

int 21h ；产生中断，

\_gets end

public \_puts

\_puts proc

mov ax,0800h ;系统调用7号功能，从键盘输入一个字符串

push ax

… ； 其他参数进栈

int 21h ；产生中断，

\_puts end

1. 利用自己设计的C库libs.obj，编写一个使用这些库函数的C语言用户程序，再编译,在与libs.obj一起链接，产生COM程序。增加内核命令执行这个程序。

main(){

char ch,str[80];

int a;

getch(&ch);

gets(str);

scanf(“a=%d”,&a);

putch(ch);

puts(str);

printint(“ch=%c, a=%d, str=%s”, ch, a, str);

}

(5)编写实验报告，描述实验工作的过程和必要的细节，如截屏或录屏，以证实实验工作的真实性

在进入程序前，栈顶安排int 21和两个字节00