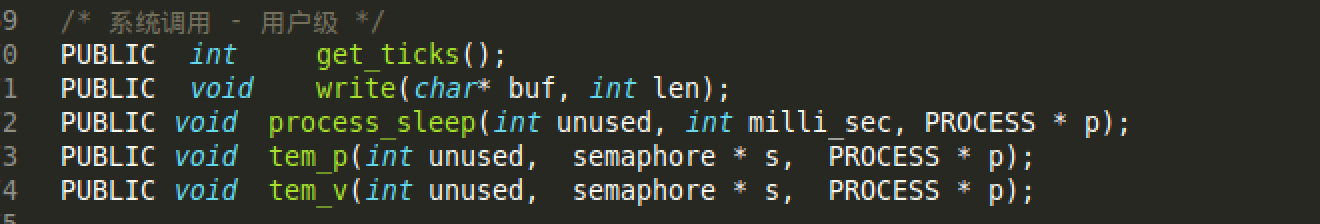
实验4补充：一开始没有用用户调用系统中断，而是直接调用系统中断，后来补上

proto.h 增加用户调用的声明



syscall.asm实现三个方法，其实就是直接调用系统中断,导出符号

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

; void process\_sleep(int unused1 int milli\_sec, PROCESS \* p)

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

process\_sleep:

mov eax, \_NR\_sleep

mov ebx, [esp + 4]

mov ecx, [esp + 8]

mov edx, [esp + 12]

int INT\_VECTOR\_SYS\_CALL

ret

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

; void tem\_p(int unused, semaphore \* s, PROCESS \* p)

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

tem\_p:

mov eax, \_NR\_p

mov ebx, [esp + 4]

mov ecx, [esp + 8]

mov edx, [esp + 12]

int INT\_VECTOR\_SYS\_CALL

ret

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

; void tem\_p(int unused, semaphore \* s, PROCESS \* p)

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

tem\_v:

mov eax, \_NR\_v

mov ebx, [esp + 4]

mov ecx, [esp + 8]

mov edx, [esp + 12]

int INT\_VECTOR\_SYS\_CALL

ret

并在global.c中增加系统调用表

