操作系统LAB4

@Author 庄子元

功能实现

1. 系统调用输出

使用系统调用printf作输出。在vsprintf里添加了对%c的解析，可以实现`printf("%c Is Reading", name)`的输出

```
case 'c':
// 解析一个char字符
    *p++ = *((char *)p_next_arg);
    p_next_arg += 4;
    break;
```

2. sleep

给每个PROCESS进程添加一个blocked和sleep_ticks属性，如果进程在睡眠就不会被调度

```
PUBLIC int sys_sleep(int milli_seconds) {
    // 设置进程睡眠
    p_proc_ready->sleep_ticks = milli_seconds;
    schedule();
    return 0;
}
```

3. PV信号量

为系统增加PV信号量

```
PUBLIC int sys_P_process(SEMAPHORE* s) {
    disable_int();
    s->value--;
    if (s->value < 0) {
        p_proc_ready->blocked = True;
        s->process_list[s->tail] = p_proc_ready;
        s->tail = (s->tail + 1) % PROCESS_LIST_SIZE;
        schedule();
    }
    enable_int();
}

PUBLIC int sys_V_process(SEMAPHORE* s) {
    disable_int();
    s->value++;
    if (s->value <= 0) {
        // 释放队列头的进程
        PROCESS* proc = s->process_list[s->head];
        proc->blocked = False;
        // 队列数组移动
```

```
            s->head = (s->head + 1) % PROCESS_LIST_SIZE;
        }
        enable_int();
    }
```

4. 修改系统栈

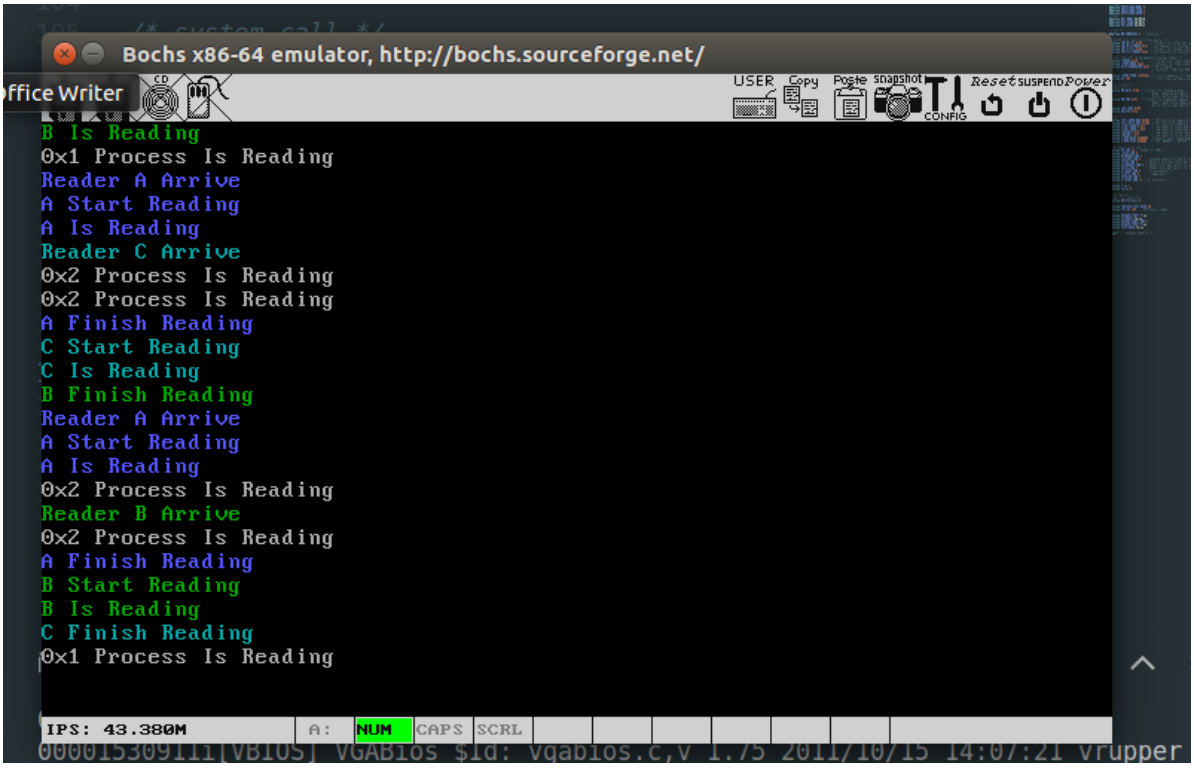
由于所有的输出都在tty0里，所以在sys_call里去掉了p_proc_ready参数的压栈

```
sys_call:
        call    save
    ; push  dword [p_proc_ready]
        sti

    push    ecx
    push    ebx
    call    [sys_call_table + eax * 4]
    add esp, 4 * 2

        mov     [esi + EAXREG - P_STACKBASE], eax
        cli
        ret
```

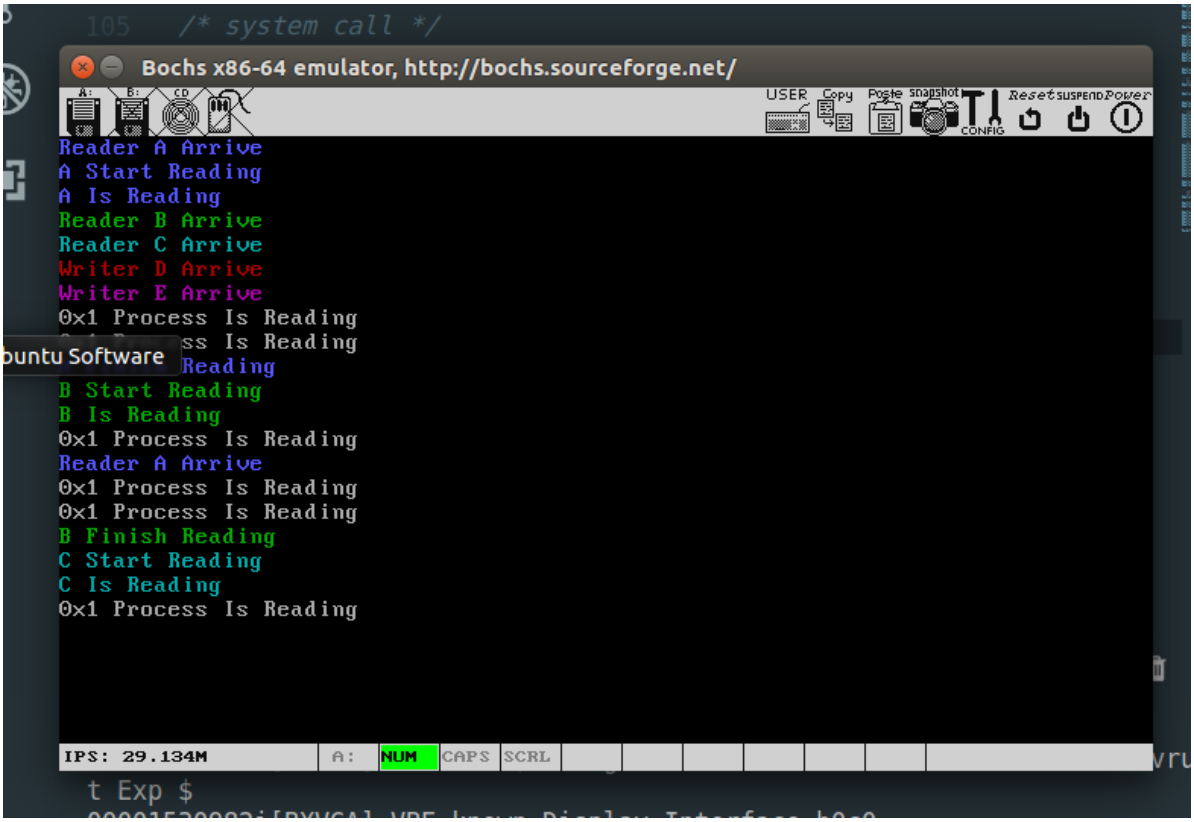
实验结果

通过修改const.h中，`READER_SAME_TIME, READER_FIRST, FAIR_READ`的属性，来选择同时可以读的用户数量，读优先、写优先、读写公平的方式。

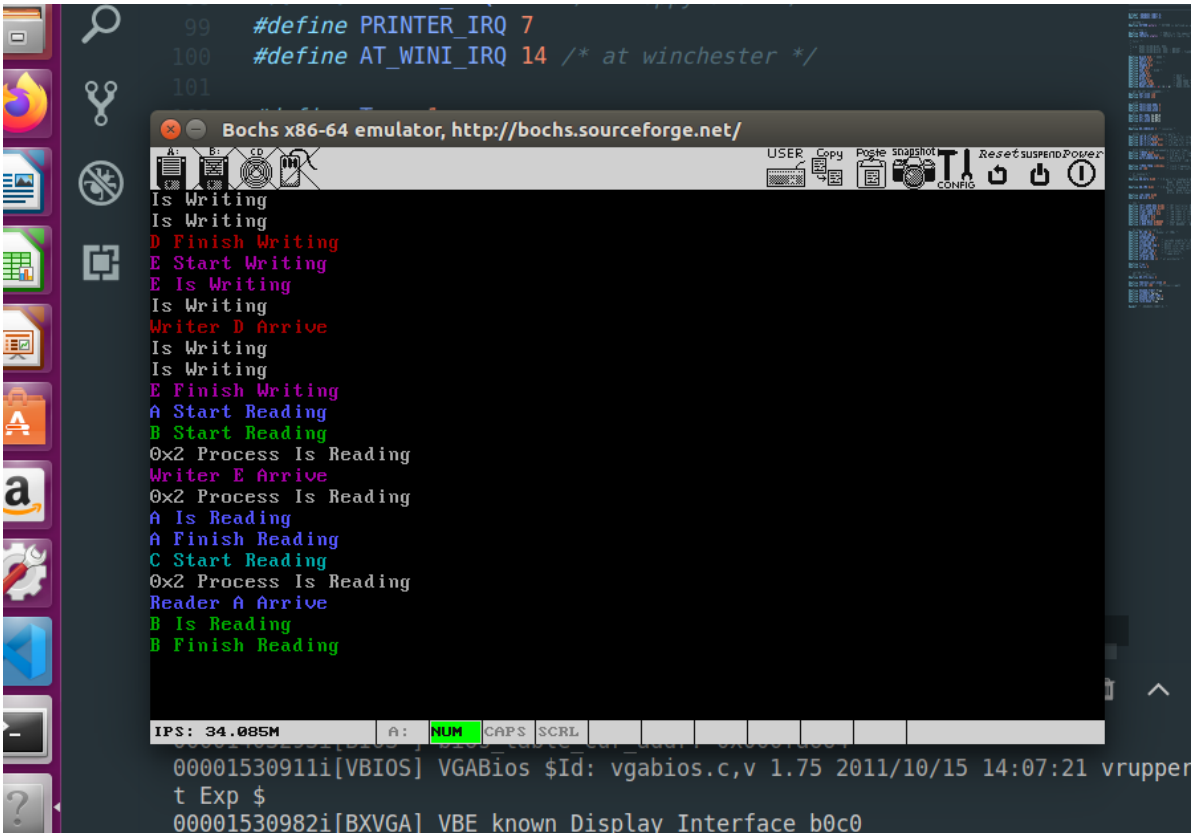
1. 读优先



两个用户可读的读优先

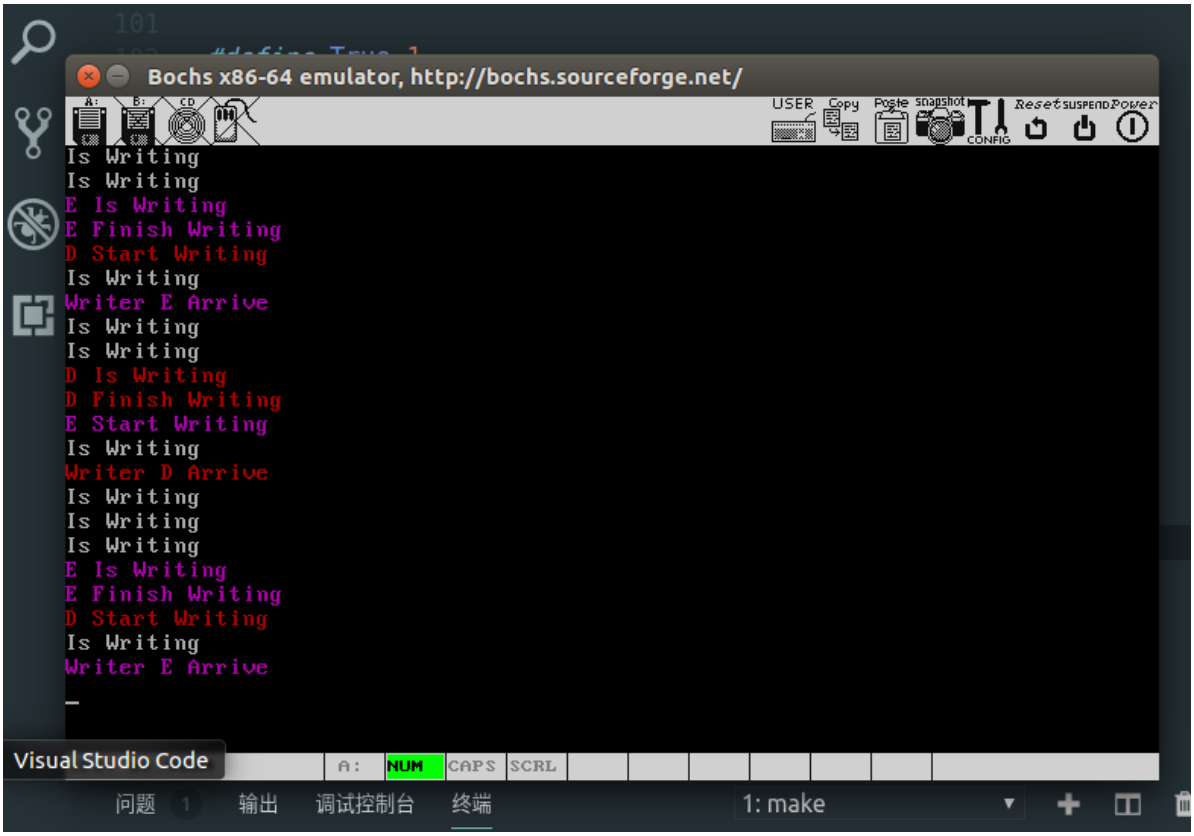


2. 读写公平



两个用户可读的读写公平

3. 写优先



两个用户可读的写优先

修改过的文件

文件夹	文件
include	console.h global.h proc.h proto.h
kernel	console.c global.c main.c proc.c vsprint.c kernel.asm syscall.asm