

Implementing our first OS service:

```
putc() //-- display a character.
```

We chose x07 as the vector number for this service. To jump to the putc() function, we use this instruction,

```
TRAP x07
```

which accesses memory location x0007, moving the content of that location into the PC. That causes the next instruction fetched to be the first instruction of the putc() service routine. The user program we wrote in class is shown at right.

Of course, the OS has to first bootup at power-on time. The PC initially contains x0200 at power-on, which we can suppose is memory-mapped to some non-volatile ROM memory containing the booter code. Let's assume the LC3 memory from x0200 to x0400 is ROM. Usually, booter code transfers the OS code from a non-volatile disk into memory. We won't write that part of our booter, we will just assume the job got done somehow. We chose the OS memory area to start at x8000. Once the OS code is in place, the booter jumps to the OS's first instruction. The booter code is shown at right.

The OS as we have written it in class is shown at right. At boot-time, the OS initializes the VT. (Here, we only initialize the x07 vector.) After initialization, the OS loads a user program and jumps to it. We will pretend all the loading has already been done: PennSim places our code at the memory locations specified by the .ORIG header for each of the three programs.

Using PennSim, assemble the three programs to get user.obj, booter.obj, and os.obj. Then, load all three .obj files into the LC3's memory. Step through program execution. The user program will call putc(). The putc() routine is incomplete. Finish it.

```
;;-----  
;;-- user.asm, user program  
;;-----  
.ORIG x1000  
  
_main:  
  
    LD R3, _char  
    TRAP x07  
  
_char: .FILL x0041  ;-- 'A'  
  
.END
```

```
;;-----  
;;-- booter.asm, OS booter  
;;-----  
.ORIG x0200  
  
;;-- goto OS:  
  
LD R7, _OS_main_address  
JMP R7  
  
_OS_main_address: .FILL x8000  
  
.END
```

```
;;-----  
;;-- os.asm, OS main program  
;;-----  
.ORIG x8000  
  
;;===== OS MAIN =====  
_main:  
  
    ;-- initialize IVT for TRAP x07 (putc):  
  
    LD R2, _07_vector_address  ;-- R2 <=== x0007  
    LD R0, _07_service_address ;-- R0 <=== &putc()  
    STR R0, R2, #0             ;-- R0 ==> MEM[ R2 ]  
  
    ;-- Goto User:  
  
    LD R7, _user_main_address  
    JMP R7  
  
_07_vector_address: .FILL x0007  
_07_service_address: .FILL _putc  
_user_main_address: .FILL x1000  
  
;;===== TRAP service routines =====  
  
;;-----  
_putc:  
;;--  
;;-- TRAP x07 service routine, display a character.  
;;-- Argument char is in R3.  
;;-----  
  
    LD R2, _DSR_address  ;-- R2 <=== &DSR  
  
    _putc_poll:          ;-- loop  
    LDR R1, R2, #0       ;-- R1 <=== DSR == MEM[ R2 ]  
    BRzp _putc_poll     ;-- until( DSR[15] == 1 )  
  
                                ;-- send ASCII to display  
  
    jmp R7                ;-- return()  
  
_DSR_address: .FILL xFE04  
  
.END
```

