Programming Project Checkpoint 1

[112006234] [Justin Thiadi 程煒財]

Task 1:

Turn in a screenshot showing compilation of your code using the provided `Makefile`. You should use the following two commands (Note: $ is the prompt displayed by the shell and is not part of the command that you type.) The first one deletes all the compiled files so it forces a rebuild if you have compiled before. The second one compiles it.

```
$ make clean
$ make
```

It should show actual compilation, warning, or error messages. Note that not all warnings are errors. The compiler should generate several `testcoop.*` files with different extensions:

- the `.hex` file can be opened directly in EdSim51
- the `.map` file shows the mapping of the symbols to their addresses after linking

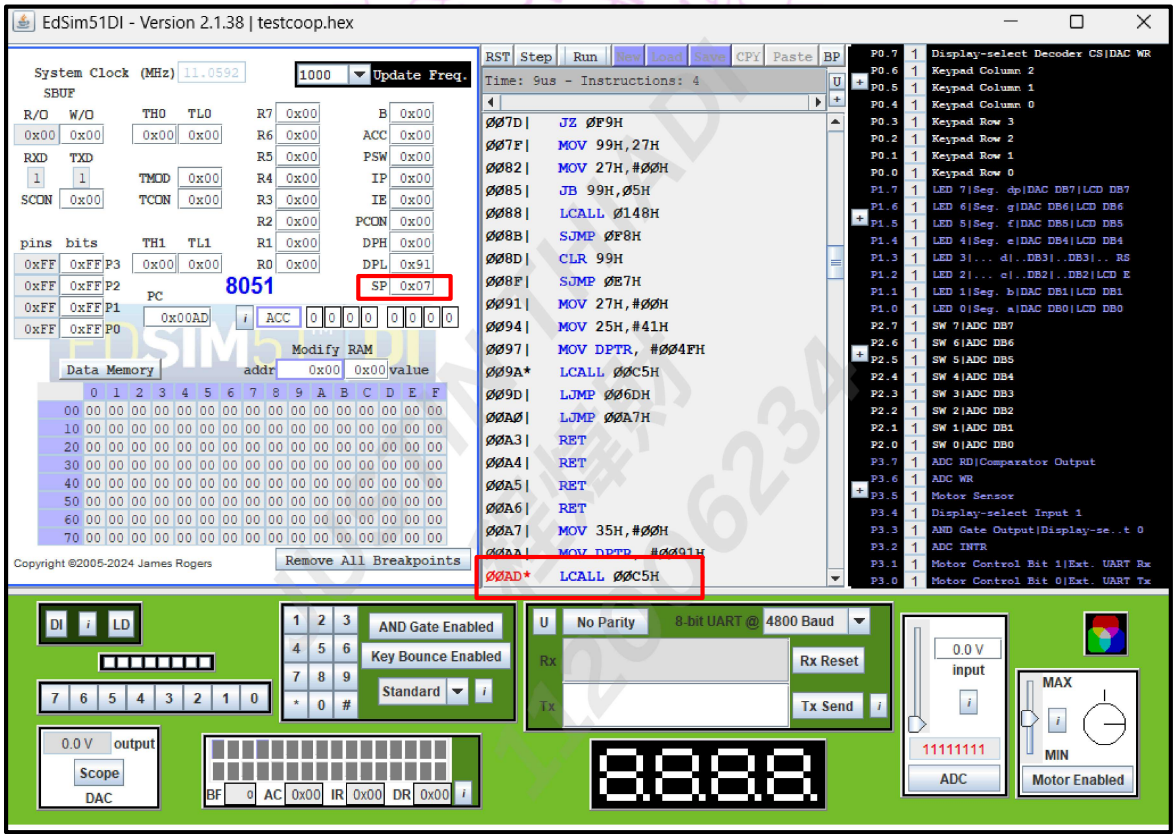
```
User@LAPTOP-U8BRQGS7 /cygdrive/C/Users/User/Documents/OS/112006234_ppc1
$ make clean
rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
rm: cannot remove '*.ihx': No such file or directory
rm: cannot remove '*.lnk': No such file or directory
make: *** [Makefile:25: clean] Error 1

User@LAPTOP-U8BRQGS7 /cygdrive/C/Users/User/Documents/OS/112006234_ppc1
$ make
sdcc -c  testcoop.c
sdcc -c  cooperative.c
cooperative.c:168: warning 85: in function ThreadCreate unreferenced function ar
gument : 'fp'
sdcc -o testcoop.hex testcoop.rel cooperative.rel

User@LAPTOP-U8BRQGS7 /cygdrive/C/Users/User/Documents/OS/112006234_ppc1
$
```

Task 2:

Look up the addresses for your symbols (i.e., functions, variables, etc) in the file `testcoop.map`. Set one or more breakpoints in EdSim51's assembly code window after you have assembled it.

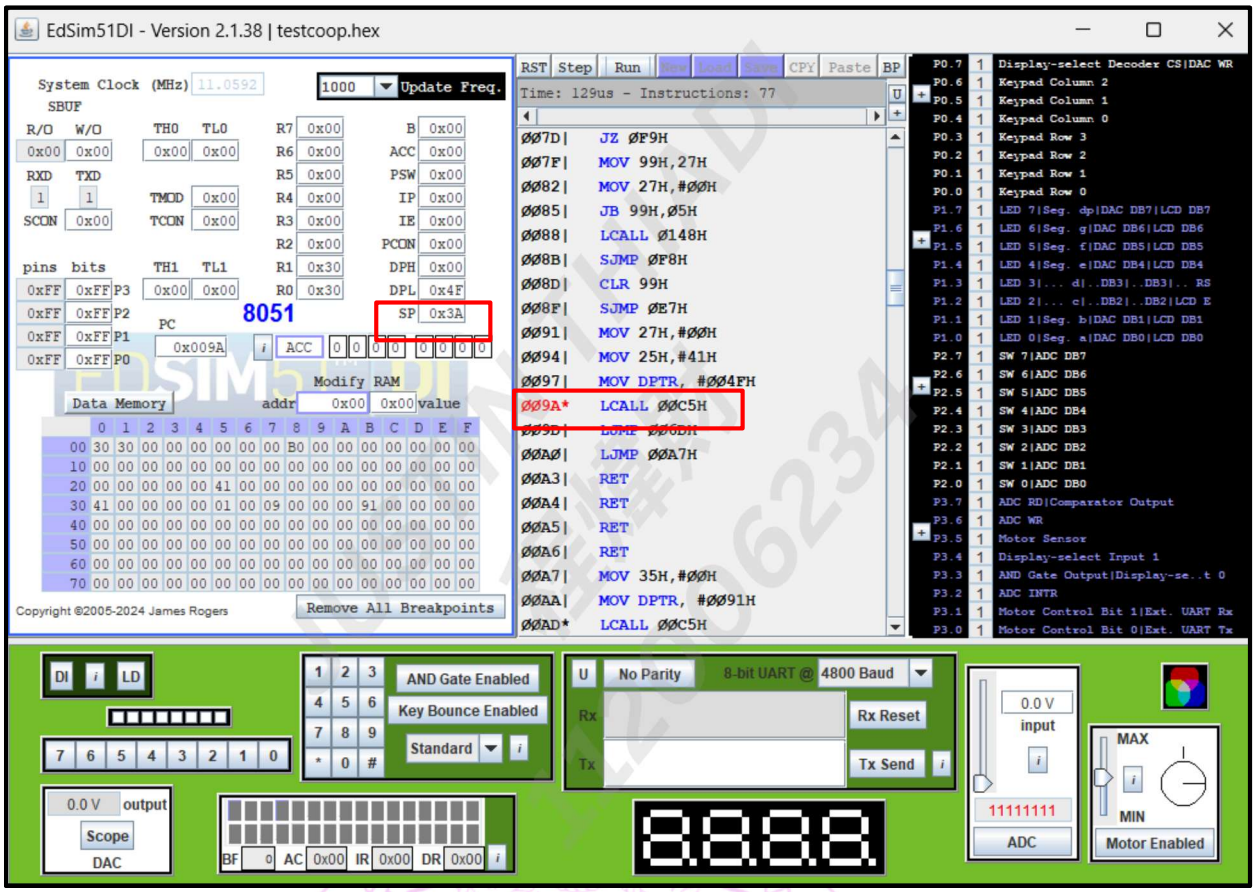
```
      Value  Global                                  Global Defined In Module
      -----  ---------------------------------------  -------------------------
C:    0000004F  _Producer                             testcoop
C:    0000006D  _Consumer                             testcoop
C:    00000091  _main                                 testcoop
C:    000000A0  __sdcc_gsinit_startup                 testcoop
C:    000000A4  __mcs51_genRAMCLEAR                   testcoop
C:    000000A5  __mcs51_genXINIT                      testcoop
C:    000000A6  __mcs51_genXRAMCLEAR                  testcoop
C:    000000A7  _Bootstrap                            cooperative
C:    000000C5  _ThreadCreate                         cooperative
C:    00000148  _ThreadYield                          cooperative
C:    0000019E  _ThreadExit                           cooperative
```

- Take one screenshot before each `ThreadCreate` call. Explain how the stack changes.



The screenshot taken above is for Thread 0 during main() execution where the PC value starts at 0xAD, then moves to 0xC5 as main() calls ThreadCreate()

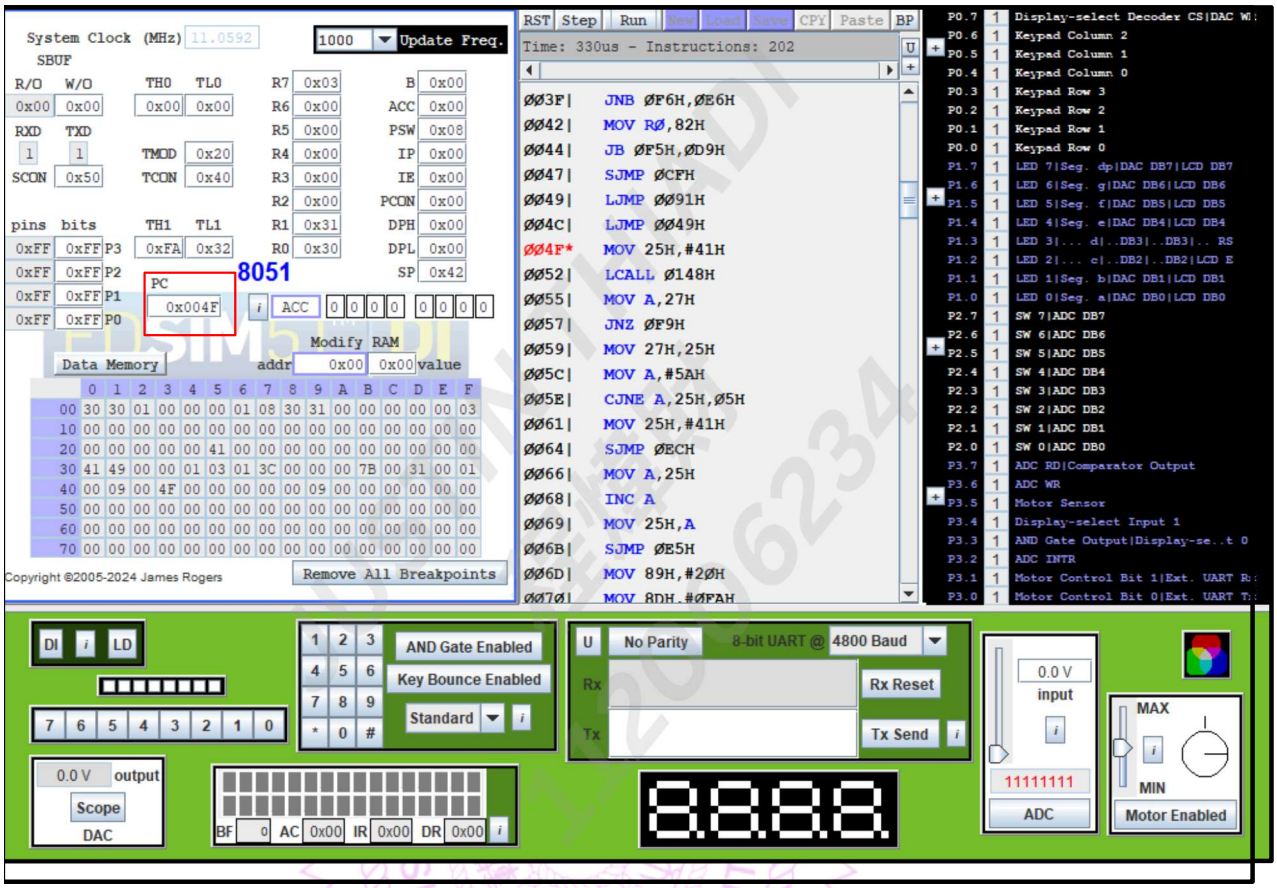
For this part, it pushes 2 bytes, then return address to main()



The screenshot taken above is for Thread 1 during simulation of ThreadCreate() for Producer where the SP value temporarily switches to 0x3A to simulate thread 1's stack

- What happens here is that:
  - Save current SP (0x3A) for main
  - Set SP = 0x3A(free area for thread 1)
  - Push 2 bytes
  - Push registers: ACC, B, DPL, DPH, PSW (5 bytes total)
  - Save SP = 0x3A to thread table (Thread 1)
  - Restore SP = 0x07 (back to thread 0) : back to main

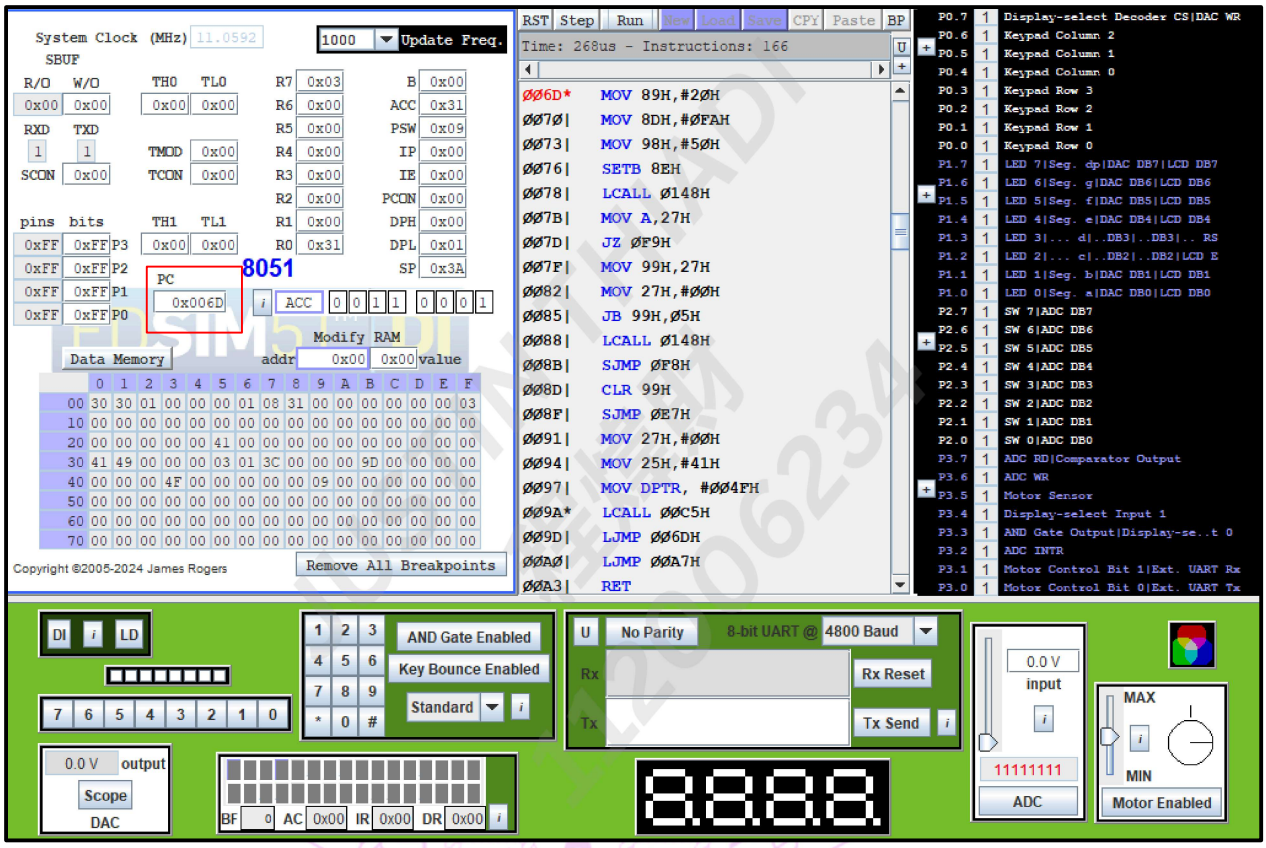
- Take one screenshot when the `Producer` is running. How do you know?



After looking up the addresses of the symbols (refer to the first screenshot of task 2), it can be seen that:

- 004F → _Producer
  - This is the exact starting address of the Producer function, so we are at the first instruction of Producer (since in the Edsim51 simulator, the current PC = 0x004F)
  - MOV 25H, #41H is the instruction at this address, which indicates an instruction for function startup — initializing variables or state
    - moving the immediate value 41H into internal RAM address 25H

- Take one screenshot when the `Consumer` is running. How do you know?



After looking up the addresses of the symbols (refer to the first screenshot of task 2), it can be seen that:

- 006D → Consumer
  - This is the exact starting address of the Consumer function, so we are at the first instruction of Consumer (since in the Edsim51 simulator, the current PC = 0x006D)
  - MOV 89H, #20H is the instruction at the address which executes the first function of the instruction, which is:
    - loading the Timer 1 low byte (TL1) with the value 20H
      - In the 8051 architecture, address 89H is part of the Special Function Register (SFR) space, which corresponds to TL1 (Timer 1 Low Byte).