

Operating System CP1

許木羽 / 111000177

1. Compiling Screenshot

a. Makefile

```
PS D:\College\OS\ppc1> mingw32-make clean
del /Q *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
PS D:\College\OS\ppc1> mingw32-make
sdcc -c testcoop.c
sdcc -c cooperative.c
cooperative.c:196: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc -o testcoop.hex testcoop.rel cooperative.rel
PS D:\College\OS\ppc1> |
```

b. .map file (function address)

Value	Global	Global Defined In Module
C: 00000009	_Producer	testcoop
C: 0000002E	_Consumer	testcoop
C: 00000054	_main	testcoop
C: 00000060	_sdcc_ginit_startup	testcoop
C: 00000064	_mcs51_genRAMCLEAR	testcoop
C: 00000065	_mcs51_genXINIT	testcoop
C: 00000066	_mcs51_genXRAMCLEAR	testcoop
C: 00000067	_Bootstrap	cooperative
C: 00000085	_ThreadCreate	cooperative
C: 000000FC	_ThreadYield	cooperative
C: 00000152	_ThreadExit	cooperative

c. .map file (variable address declared by me)

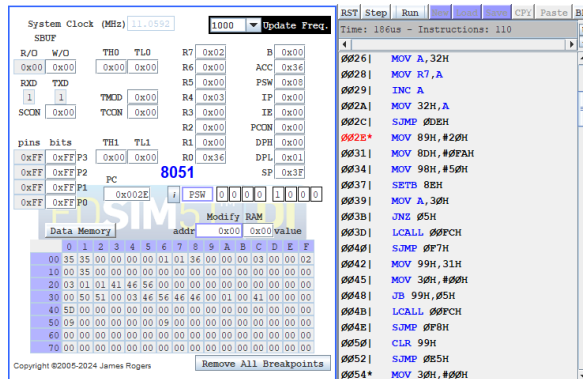
Value	Global	Global Defined In Module
00000000	_.ABS.	cooperative
00000030	_available	testcoop
00000031	_globalBuffer	testcoop
00000032	_producerChar	testcoop
00000033	_currentThreadID	cooperative
00000034	_bitmap	cooperative
00000035	_savedStackPointers	cooperative
0000003A	_newThreadID	cooperative
0000003B	_i	cooperative
0000003C	_currentSP	cooperative

2. Screenshot of EdSim51

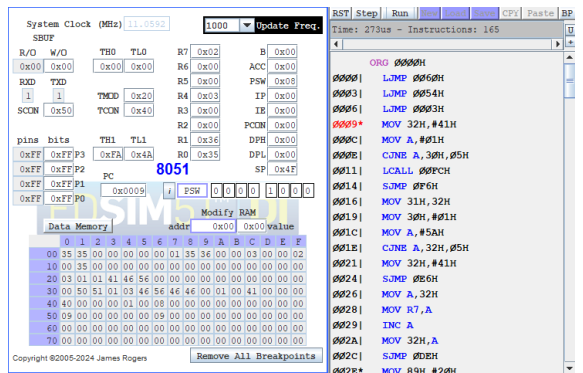
a. Create Main Thread

The screenshot displays the EdSim51 software interface. The top panel shows the System Clock at 11.0592 MHz and the Update Freq. set to 1000. The Register File window shows the current state of the processor registers. The Program Counter (PC) is set to 8051. The Data Memory window shows the current state of the data memory. The assembly code window shows the instructions for the main thread creation process. The CPU window shows the current state of the processor.

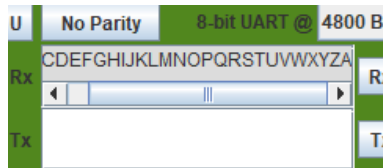
b. Call Consumer Function



c. Create Producer Thread



d. Output





















3. Explanation

2 functions run simultaneously, consumer and producer in a different thread. Producer will wait until the global buffer is not available, else it will yield. The global buffer will be set value by the producer and then set the flag of available = true. While consumer wait until the global buffer is available, then it will take to SBUF and set available to false. There should be no issue for them sharing the global buffer since they set the availability after setting up the global buffer or taking the global buffer.

For the code, I follow every instruction step by step. In this checkpoint, I use memory around 0x3X to declare my own variable. The Bitmap method, I use range 0x0-0xF to determine 4 threads, each bit for own thread validation. As the base starts at 0x3F. Once a thread is called, it changes to 0x4F as the size of each thread is 0x10. Since only 2 Threads were created, the current SP will be 0x4F.

I know when the Producer and Consumer running by marking a breakpoint inside the EdSim51 code, for the address I know by saw the .map file and see the address of the functions.

 cooperative.asm Type: ASM File
 cooperative.c Type: C Source File
 cooperative.h Type: C Header Source File
 cooperative.lst Type: LST File
 cooperative.rel Type: REL File
 cooperative.rst Type: Restructured Text Source File
 cooperative.sym Type: SYM File
 Makefile Type: File
 testcoop.asm Type: ASM File
 testcoop.c Type: C Source File
 testcoop.hex Type: HEX File
 testcoop.lk Type: LK File
 testcoop.lst Type: LST File
 testcoop.map Type: MAP File
 testcoop.mem Type: MEM File
 testcoop.rel Type: REL File
 testcoop.rst Type: Restructured Text Source File
 testcoop.sym Type: SYM File