107062381 楊孝偉

Checkpoint 1 Report

Testcoop.c

In this file, I will use only two variables : buffer and token. Buffer is used for the shared buffer between the producer and consumer. Token is used to generate characters from A to Z for the producer.

In the main function, buffer is initialized to ‘1”, which simply means that buffer is empty and producer can generate an alphabet to it. Token is initialized to A. Main will spawn consumer thread, while main itself calls producer. This is done to not waste main’s thread, more economical.

In the producer function, before it enters the infinite loop, I initialized the token to ‘A’ first, to make sure that token is really initialized to ‘A’. After it enters the loop, it will check whether buffer is not equal to 1, if yes, then buffer is still full (consumer hasn’t consumed yet), producer will thread yield. Else if buffer is already empty, buffer will take the value in token, then update the value in token.

In the consumer function, before it enters the infinite loop, we initialize Tx for polling. After it enters the loop, it will check whether buffer is equal to 1, if yes, then buffer is still empty (producer hasn’t produced it yet), then it will thread yield. Else, SBUF(reading or writing register) will take the value in buffer, then we reset buffer to 1 to indicate if it’s empty, then we check if Tx is busy (serial port hasn’t finished writing it yet), then we thread yield again. Finally, we reset It to 0 again.

Cooperative.c

In this file, I use quite a lot of variables, I will explain some later as I explain the functions. Sp[4] is used for the saved stack pointers. Cur_thread is for the current thread ID. Bitmap is used to determine which thread ID is a valid thread.

First, we define SAVESTATE. It is a C macro for saving the context of the current thread and it is written in inlined assembly. First we push the ACC, B register, DPTR, and PSW onto stack. Then we save the stack pointer for the current thread into the saved stack pointers array as indexed by the current thread ID.

Next, we define a C macro for restoring the context of the current thread by basically doing reverse the operation of SAVESTATE.

Bootstrap is the start-up code to set up and run the first thread. First we initialize the bitmap to 0000 (which is 0 in decimal value), which means that all of the threads are available for use. Then we create a thread for main and set current thread to this thread ID and restore its context with RESTORESTATE so that it starts running the main function. Stack was set up by the threadcreate(main). Stack 0 now has the return address of main.

Next is the thread create function to create a thread data structure so it is ready to be restored. First we check to see whether we have not reached the maximum number of threads. Maximum number of threads is reached if the bitmap value is equal to 15 (which is 1111 in binary value), because we will set the bit value to 1 if a thread is used. If so, we will return -1 which is not a valid thread ID. Otherwise, we will find a thread ID that is not in use and grab it. I did this by first initializing a variable called mark to 1 (this means that mark will have the value of 0001 in binary initially). Then it will enter the while loop.

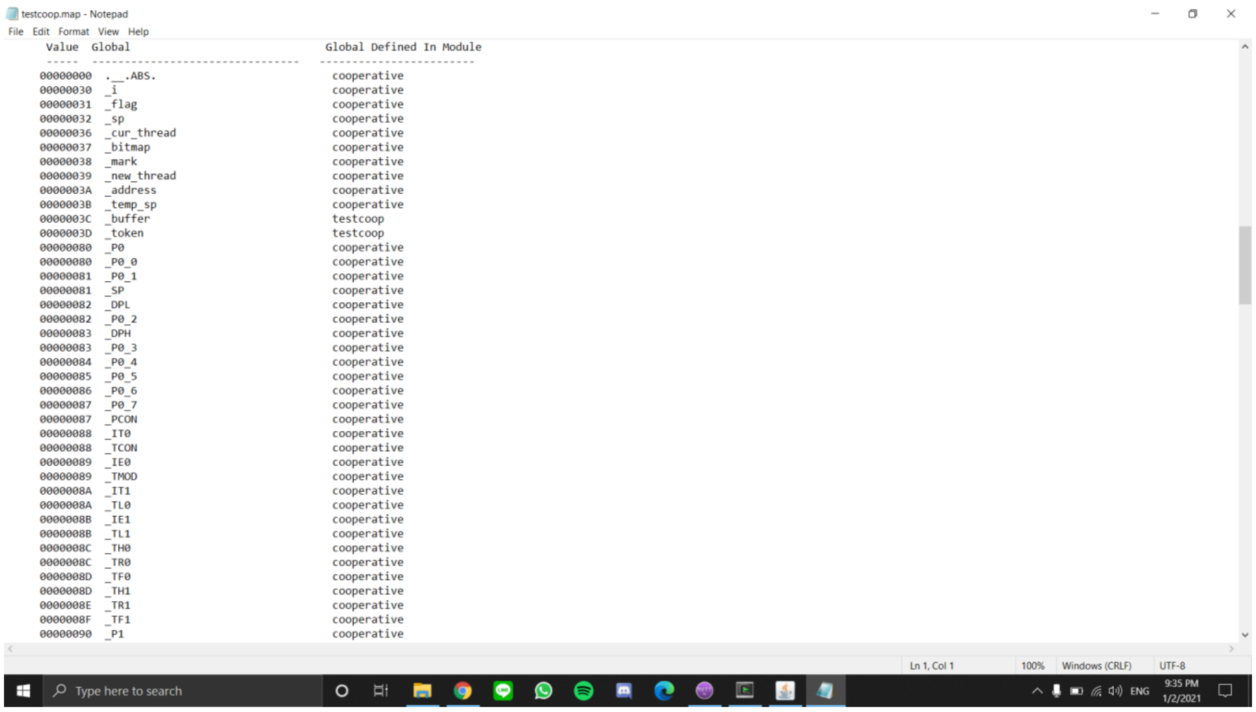
107062381 楊孝偉

First it will look for which thread is available by keep shifting the mark to the left by 1 bit position. If there is an available thread, then bitmap & mark value will be 0, then we simply set the new thread accordingly. Then we update the bitmap value to indicate that the selected thread is now used by XOR-ing the bitmap with mark. Then we calculate the starting stack location for new thread. We will check if new thread is 0, 1, 2 or 3. If new thread is 0, we set the address to 0x3F as the hardware stack in 8051 is pre increment. If new thread is 1, 2 and 3, we set the address to 0x4F, 0x5F, 0x6F respectively. Then we save the current SP in a temporary and set SP to the starting location for the new thread, which is address basically. Then we push the return address of fp onto stack by pushing DPL and DPH as in SDCC convention, 2-byte ptr is passed in DPTR, but push instruction can only push it as two separate registers, DPL and DPH. Then we initialize the registers to 0. So we assign ACC to 0 by ANL, an instruction to perform a bitwise logical AND operation between the specified operands and stores the result in the destination operand (ACC), and push it four times for ACC, B, DPL and DPH. Then we need to push PSW registers. RS1 and RS0 is used for register bank selection. Therefore, we can fill RS1 and RS0 by shifting the new thread ID by 3 as RS1 and RS0 is 4 and 3 bits away from the least significant bit. Then we push the PSW. Then we can write the current stack pointer to the saved stack pointer array, SP will take the previously saved SP, and return this newly created thread ID.

Thread Yield function is called by a running thread to yield control to the other thread. First it will find the next thread that can run and set the current thread ID to it. Then if the bitwise AND value of bitmap and the current thread ID is not equal to 0, we break from the while loop. This means that thread can run, so we break from the loop. If the bitwise AND value of that thread ID and bitmask is equal to 0, this means that thread ID is still not active, then the while loop will continue to find the next thread that can run. It is guaranteed that at least one thread is active, so this loop will always terminate.

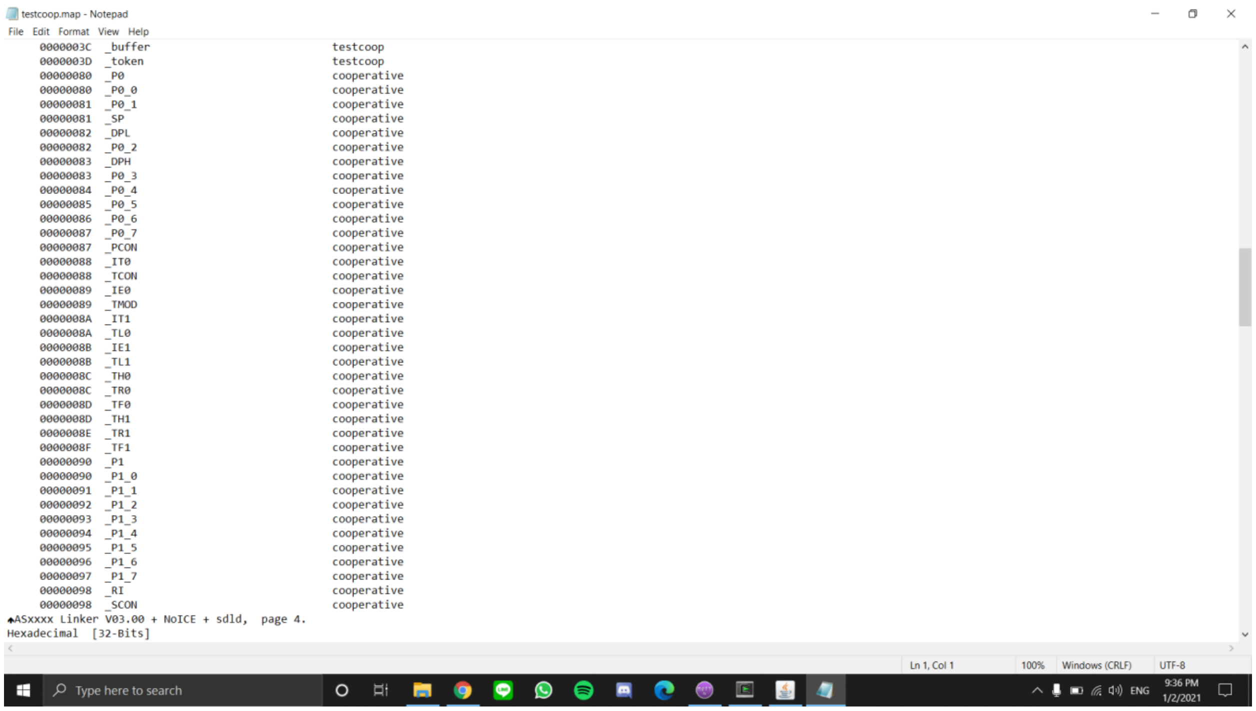
I did not fill the Thread Exit function as this function is not used anywhere.

Testcoop.map

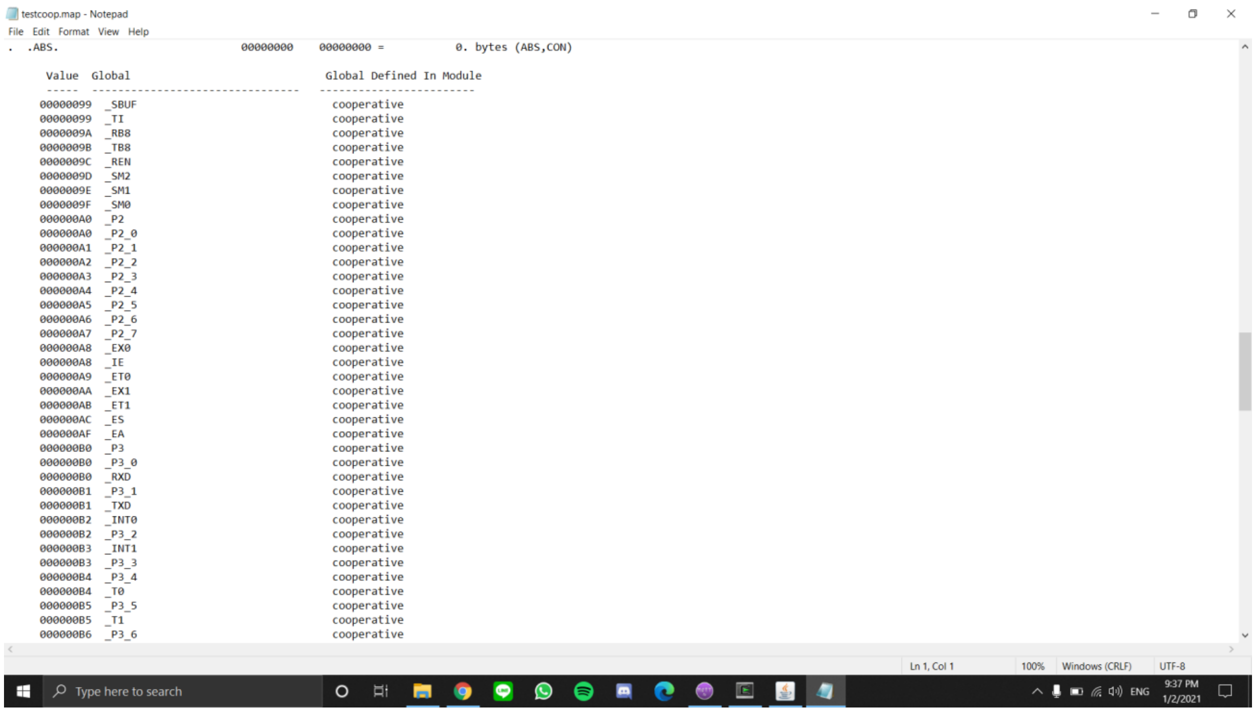


```
     Value  Global                              Global Defined In Module
     -----  --------------------------------   ------------------------
    00000000  .__.ABS.                           cooperative
    00000030  _i                                 cooperative
    00000031  _flag                              cooperative
    00000032  _sp                                cooperative
    00000036  _cur_thread                        cooperative
    00000037  _bitmap                            cooperative
    00000038  _mark                              cooperative
    00000039  _new_thread                        cooperative
    0000003A  _address                           cooperative
    0000003B  _temp_sp                           cooperative
    0000003C  _buffer                            testcoop
    0000003D  _token                             testcoop
    00000080  _P0                                cooperative
    00000080  _P0_0                              cooperative
    00000081  _P0_1                              cooperative
    00000081  _SP                                cooperative
    00000082  _DPL                               cooperative
    00000082  _P0_2                              cooperative
    00000083  _DPH                               cooperative
    00000083  _P0_3                              cooperative
    00000084  _P0_4                              cooperative
    00000085  _P0_5                              cooperative
    00000086  _P0_6                              cooperative
    00000087  _P0_7                              cooperative
    00000087  _PCON                              cooperative
    00000088  _IT0                               cooperative
    00000088  _TCON                              cooperative
    00000089  _IE0                               cooperative
    00000089  _TMOD                              cooperative
    0000008A  _IT1                               cooperative
    0000008A  _TL0                               cooperative
    0000008B  _IE1                               cooperative
    0000008B  _TL1                               cooperative
    0000008C  _TH0                               cooperative
    0000008C  _TR0                               cooperative
    0000008D  _TF0                               cooperative
    0000008D  _TH1                               cooperative
    0000008E  _TR1                               cooperative
    0000008F  _TF1                               cooperative
    00000090  _P1                                cooperative
```

107062381 楊孝偉

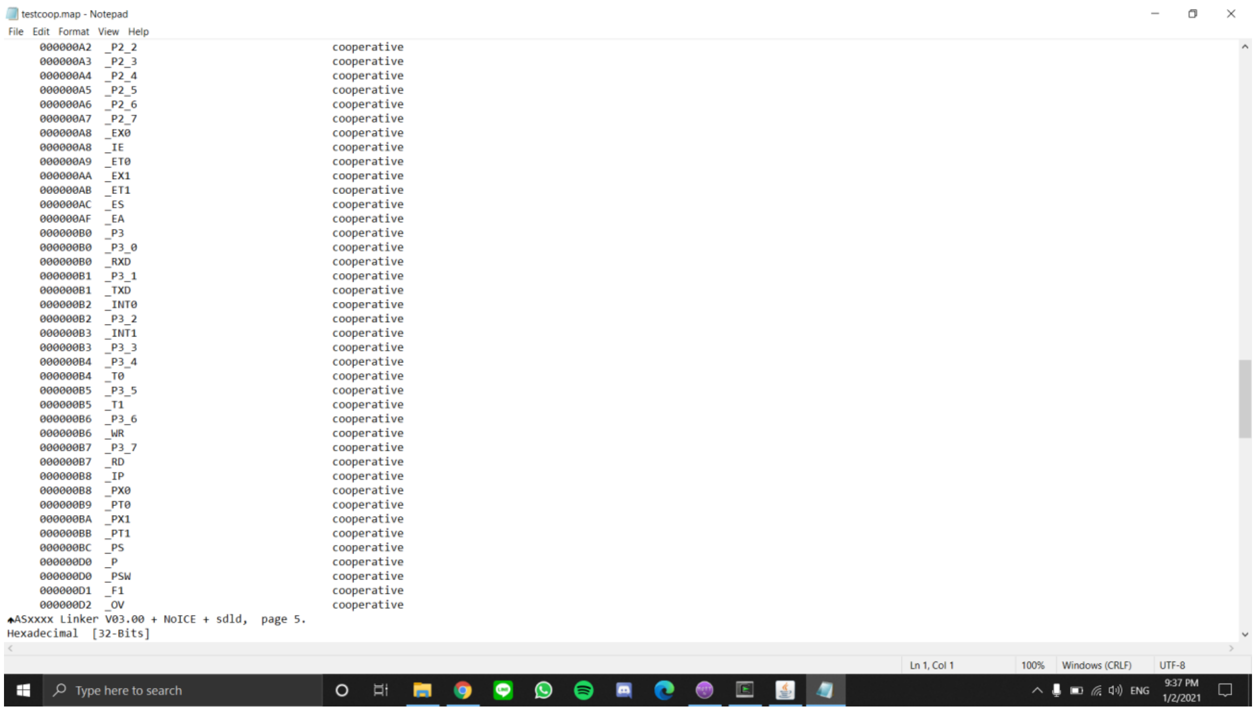


```
     0000003C  _buffer                           testcoop
     0000003D  _token                            testcoop
     00000080  _P0                               cooperative
     00000080  _P0_0                             cooperative
     00000081  _P0_1                             cooperative
     00000081  _SP                               cooperative
     00000082  _DPL                              cooperative
     00000082  _P0_2                             cooperative
     00000083  _DPH                              cooperative
     00000083  _P0_3                             cooperative
     00000084  _P0_4                             cooperative
     00000085  _P0_5                             cooperative
     00000086  _P0_6                             cooperative
     00000087  _P0_7                             cooperative
     00000087  _PCON                             cooperative
     00000088  _IT0                              cooperative
     00000088  _TCON                             cooperative
     00000089  _IE0                              cooperative
     00000089  _TMOD                             cooperative
     0000008A  _IT1                              cooperative
     0000008A  _TL0                              cooperative
     0000008B  _IE1                              cooperative
     0000008B  _TL1                              cooperative
     0000008C  _TH0                              cooperative
     0000008C  _TR0                              cooperative
     0000008D  _TF0                              cooperative
     0000008D  _TH1                              cooperative
     0000008E  _TR1                              cooperative
     0000008F  _TF1                              cooperative
     00000090  _P1                               cooperative
     00000090  _P1_0                             cooperative
     00000091  _P1_1                             cooperative
     00000092  _P1_2                             cooperative
     00000093  _P1_3                             cooperative
     00000094  _P1_4                             cooperative
     00000095  _P1_5                             cooperative
     00000096  _P1_6                             cooperative
     00000097  _P1_7                             cooperative
     00000098  _RI                               cooperative
     00000098  _SCON                             cooperative
ASxxxx Linker V03.00 + NoICE + sdld,  page 4.
Hexadecimal  [32-Bits]
```

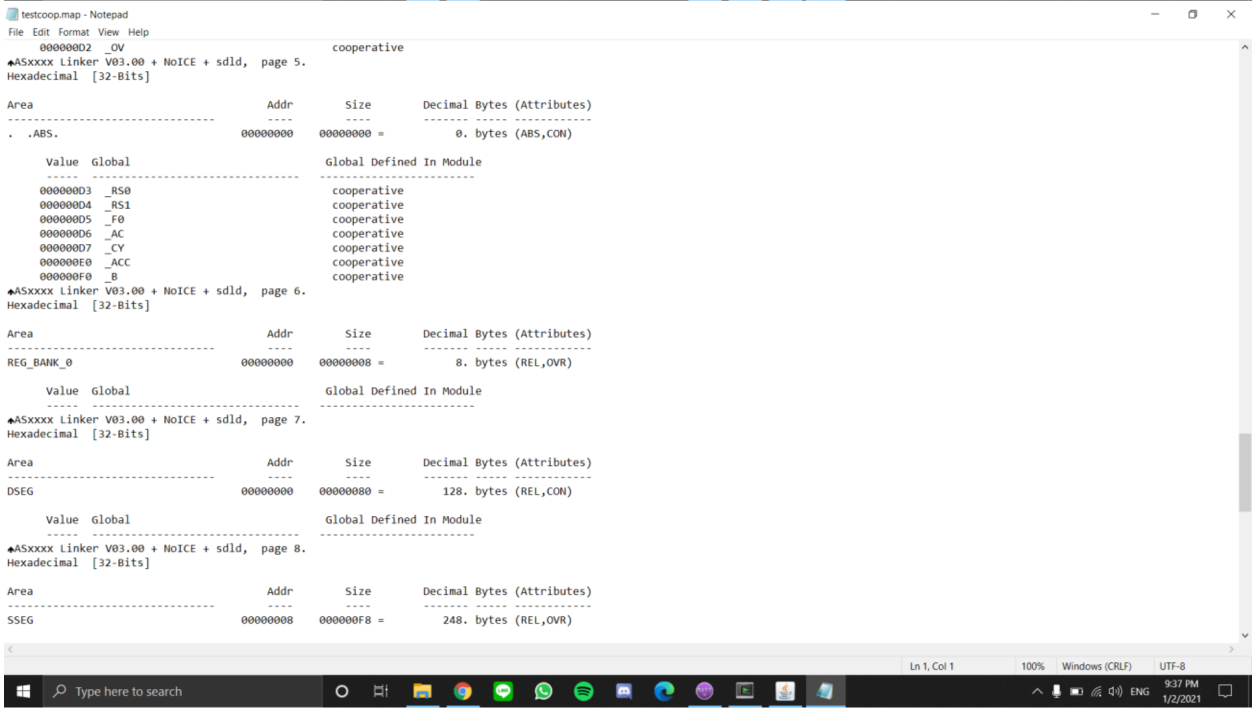


```
.  .ABS.                              00000000    00000000 =           0. bytes (ABS,CON)

      Value  Global                              Global Defined In Module
      -----  --------------------------------   ------------------------
     00000099  _SBUF                             cooperative
     00000099  _TI                               cooperative
     0000009A  _RB8                              cooperative
     0000009B  _TB8                              cooperative
     0000009C  _REN                              cooperative
     0000009D  _SM2                              cooperative
     0000009E  _SM1                              cooperative
     0000009F  _SM0                              cooperative
     000000A0  _P2                               cooperative
     000000A0  _P2_0                             cooperative
     000000A1  _P2_1                             cooperative
     000000A2  _P2_2                             cooperative
     000000A3  _P2_3                             cooperative
     000000A4  _P2_4                             cooperative
     000000A5  _P2_5                             cooperative
     000000A6  _P2_6                             cooperative
     000000A7  _P2_7                             cooperative
     000000A8  _EX0                              cooperative
     000000A8  _IE                               cooperative
     000000A9  _ET0                              cooperative
     000000AA  _EX1                              cooperative
     000000AB  _ET1                              cooperative
     000000AC  _ES                               cooperative
     000000AF  _EA                               cooperative
     000000B0  _P3                               cooperative
     000000B0  _P3_0                             cooperative
     000000B0  _RXD                              cooperative
     000000B1  _P3_1                             cooperative
     000000B1  _TXD                              cooperative
     000000B2  _INT0                             cooperative
     000000B2  _P3_2                             cooperative
     000000B3  _INT1                             cooperative
     000000B3  _P3_3                             cooperative
     000000B4  _P3_4                             cooperative
     000000B4  _T0                               cooperative
     000000B5  _P3_5                             cooperative
     000000B5  _T1                               cooperative
     000000B6  _P3_6                             cooperative
```

107062381 楊孝偉

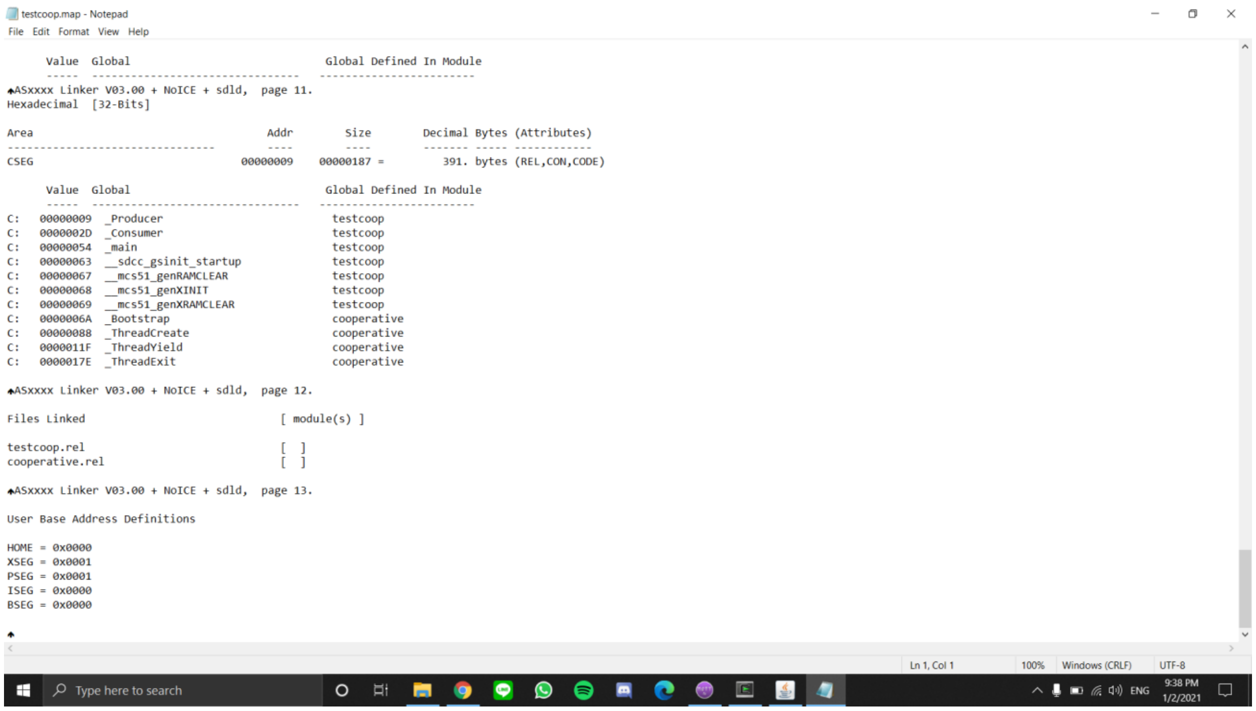


```
     000000A2  _P2_2                            cooperative
     000000A3  _P2_3                            cooperative
     000000A4  _P2_4                            cooperative
     000000A5  _P2_5                            cooperative
     000000A6  _P2_6                            cooperative
     000000A7  _P2_7                            cooperative
     000000A8  _EX0                             cooperative
     000000A8  _IE                              cooperative
     000000A9  _ET0                             cooperative
     000000AA  _EX1                             cooperative
     000000AB  _ET1                             cooperative
     000000AC  _ES                              cooperative
     000000AF  _EA                              cooperative
     000000B0  _P3                              cooperative
     000000B0  _P3_0                            cooperative
     000000B0  _RXD                             cooperative
     000000B1  _P3_1                            cooperative
     000000B1  _TXD                             cooperative
     000000B2  _INT0                            cooperative
     000000B2  _P3_2                            cooperative
     000000B3  _INT1                            cooperative
     000000B3  _P3_3                            cooperative
     000000B4  _P3_4                            cooperative
     000000B4  _T0                              cooperative
     000000B5  _P3_5                            cooperative
     000000B5  _T1                              cooperative
     000000B6  _P3_6                            cooperative
     000000B6  _WR                              cooperative
     000000B7  _P3_7                            cooperative
     000000B7  _RD                              cooperative
     000000B8  _IP                              cooperative
     000000B8  _PX0                             cooperative
     000000B9  _PT0                             cooperative
     000000BA  _PX1                             cooperative
     000000BB  _PT1                             cooperative
     000000BC  _PS                              cooperative
     000000D0  _P                               cooperative
     000000D0  _PSW                             cooperative
     000000D1  _F1                              cooperative
     000000D2  _OV                              cooperative
♠ASxxxx Linker V03.00 + NoICE + sdld,  page 5.
Hexadecimal  [32-Bits]
```



```
     000000D2  _OV                              cooperative
♠ASxxxx Linker V03.00 + NoICE + sdld,  page 5.
Hexadecimal  [32-Bits]

Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------        ----        ----        ------- ----- ------------
.  .ABS.                            00000000    00000000 =           0. bytes (ABS,CON)

      Value  Global                              Global Defined In Module
      -----  --------------------------------    ------------------------
     000000D3  _RS0                             cooperative
     000000D4  _RS1                             cooperative
     000000D5  _F0                              cooperative
     000000D6  _AC                              cooperative
     000000D7  _CY                              cooperative
     000000E0  _ACC                             cooperative
     000000F0  _B                               cooperative
♠ASxxxx Linker V03.00 + NoICE + sdld,  page 6.
Hexadecimal  [32-Bits]

Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------        ----        ----        ------- ----- ------------
REG_BANK_0                          00000000    00000008 =           8. bytes (REL,OVR)

      Value  Global                              Global Defined In Module
      -----  --------------------------------    ------------------------
♠ASxxxx Linker V03.00 + NoICE + sdld,  page 7.
Hexadecimal  [32-Bits]

Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------        ----        ----        ------- ----- ------------
DSEG                                00000000    00000080 =         128. bytes (REL,CON)

      Value  Global                              Global Defined In Module
      -----  --------------------------------    ------------------------
♠ASxxxx Linker V03.00 + NoICE + sdld,  page 8.
Hexadecimal  [32-Bits]

Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------        ----        ----        ------- ----- ------------
SSEG                                00000008    000000F8 =         248. bytes (REL,OVR)
```

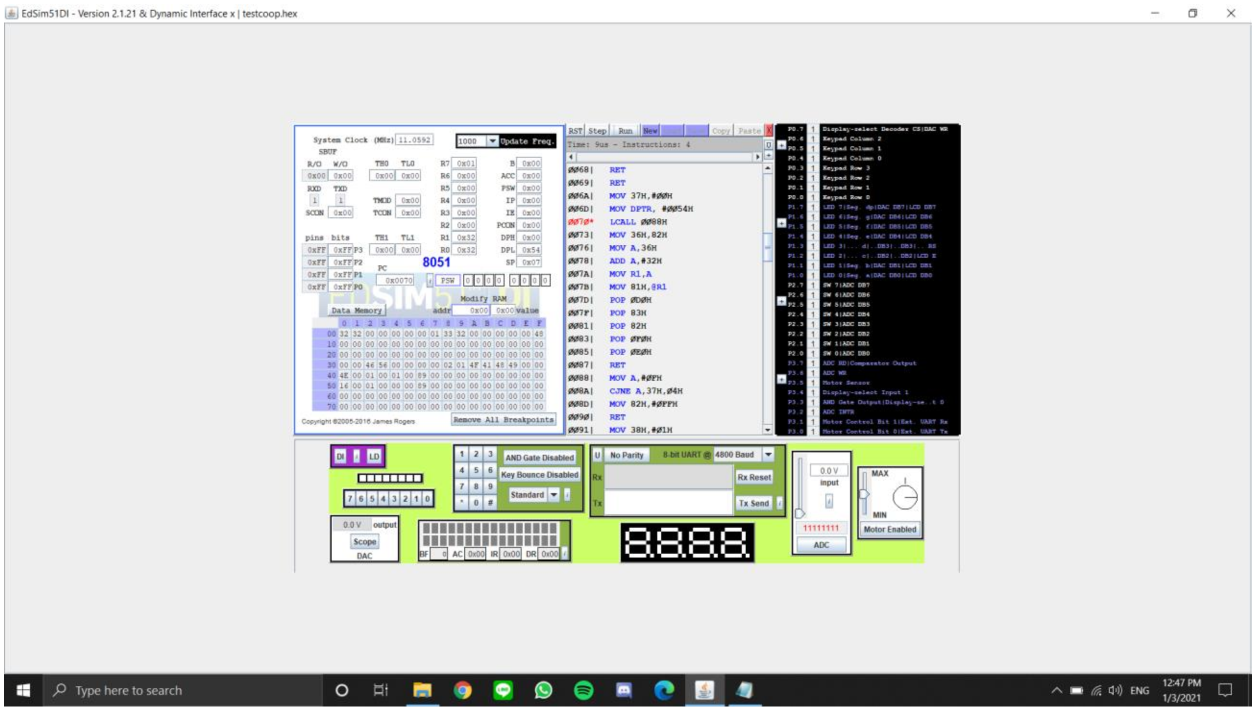
107062381 楊孝偉



```
      Value  Global                              Global Defined In Module
      -----  --------------------------------   ------------------------
ASxxxx Linker V03.00 + NoICE + sdld,  page 11.
Hexadecimal  [32-Bits]

Area                                    Addr        Size        Decimal Bytes (Attributes)
--------------------------------        ----        ----        ------- ----- ------------
CSEG                                00000009    00000187 =        391. bytes (REL,CON,CODE)

      Value  Global                              Global Defined In Module
      -----  --------------------------------   ------------------------
C:   00000009  _Producer                          testcoop
C:   0000002D  _Consumer                          testcoop
C:   00000054  _main                              testcoop
C:   00000063  __sdcc_gsinit_startup              testcoop
C:   00000067  __mcs51_genRAMCLEAR                testcoop
C:   00000068  __mcs51_genXINIT                   testcoop
C:   00000069  __mcs51_genXRAMCLEAR               testcoop
C:   0000006A  _Bootstrap                         cooperative
C:   00000088  _ThreadCreate                      cooperative
C:   0000011F  _ThreadYield                       cooperative
C:   0000017E  _ThreadExit                        cooperative

ASxxxx Linker V03.00 + NoICE + sdld,  page 12.

Files Linked                              [ module(s) ]

testcoop.rel                              [  ]
cooperative.rel                           [  ]

ASxxxx Linker V03.00 + NoICE + sdld,  page 13.

User Base Address Definitions

HOME = 0x0000
XSEG = 0x0001
PSEG = 0x0001
ISEG = 0x0000
BSEG = 0x0000
```

Breakpoints

1. Threadcreate(main)
   Before calling threadcreate(main)

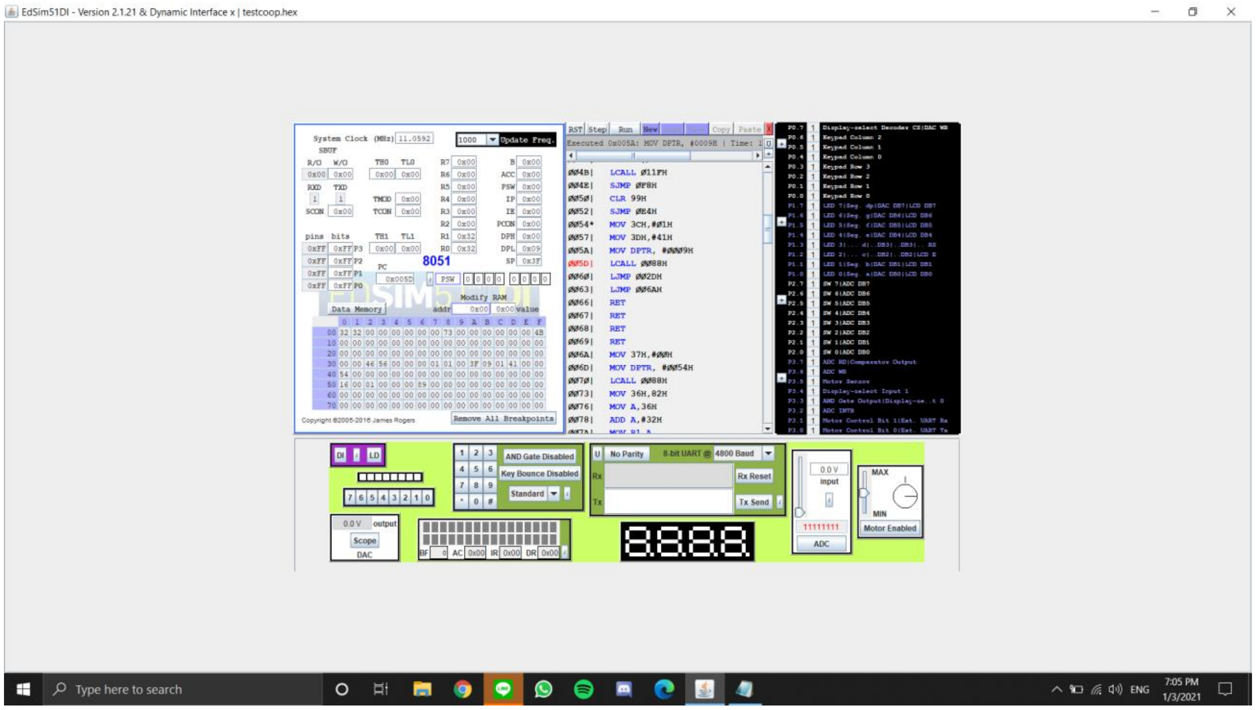


```
System Clock (MHz) 11.0592        1000  Update Freq.
SBUF
R/O   W/O     TH0   TL0     R7 0x01    B    0x00
0x00  0x00    0x00  0x00    R6 0x00    ACC  0x00
RXD   TXD                   R5 0x00    PSW  0x00
1     1       TMOD  0x00    R4 0x00    IP   0x00
SCON  0x00    TCON  0x00    R3 0x00    IE   0x00
                            R2 0x00    PCON 0x00
pins  bits    TH1   TL1     R1 0x32    DPH  0x00
0xFF  0xFF P3 0x00  0x00    R0 0x32    DPL  0x54
0xFF  0xFF P2                          SP   0x07
0xFF  0xFF P1  PC 0x0070    PSW 0 0 0 0 0 0 0 0
0xFF  0xFF P0

     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  32 32 00 00 00 00 00 01 33 32 00 00 00 00 00 48
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30  00 00 46 56 00 00 00 00 02 01 4F 41 48 49 00 00
40  4E 00 01 00 01 00 89 00 00 00 00 00 00 00 00 00
50  16 00 01 00 00 00 89 00 00 00 00 00 00 00 00 00
60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Time: 9us - Instructions: 4
ØØ68|  RET
ØØ69|  RET
ØØ6A|  MOV 37H,#ØØH
ØØ6D|  MOV DPTR, #ØØ54H
ØØ7Ø*  LCALL ØØ88H
ØØ73|  MOV 36H,82H
ØØ76|  MOV A,36H
ØØ78|  ADD A,#32H
ØØ7A|  MOV R1,A
ØØ7B|  MOV 81H,@R1
ØØ7D|  POP ØDØH
ØØ7F|  POP 83H
ØØ81|  POP 82H
ØØ83|  POP ØFØH
ØØ85|  POP ØEØH
ØØ87|  RET
ØØ88|  MOV A,#ØFH
ØØ8A|  CJNE A,37H,Ø4H
ØØ8D|  MOV 82H,#ØFFH
ØØ9Ø|  RET
ØØ91|  MOV 38H,#Ø1H
```

107062381 楊孝偉

Before calling threadcreate(producer)



EdSim51DI - Version 2.1.21 & Dynamic Interface x | testcoop.hex

```
System Clock (MHz) 11.0592        1000  Update Freq.

R7 0x00   B    0x00
R6 0x00   ACC  0x00
R5 0x00   PSW  0x00
R4 0x00   IP   0x00
R3 0x00   IE   0x00
R2 0x00   PCON 0x00
R1 0x32   DPH  0x00
R0 0x32   DPL  0x09
          SP   0x3F
PC 0x005D

Data Memory
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  32 32 00 00 00 00 00 00 73 00 00 00 00 00 00 4B
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30  00 00 46 56 00 00 00 01 01 00 3F 09 01 41 00 00
40  54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50  16 00 01 00 00 00 89 00 00 00 00 00 00 00 00 00
60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Executed 0x005A: MOV DPTR, #0009H | Time: 1
ØØ4B|   LCALL ØllFH
ØØ4E|   SJMP ØF8H
ØØ5Ø|   CLR 99H
ØØ52|   SJMP ØE4H
ØØ54*   MOV 3CH,#Ø1H
ØØ57|   MOV 3DH,#41H
ØØ5A|   MOV DPTR, #ØØØ9H
ØØ5D|   LCALL ØØ88H
ØØ6Ø|   LJMP ØØ2DH
ØØ63|   LJMP ØØ6AH
ØØ66|   RET
ØØ67|   RET
ØØ68|   RET
ØØ69|   RET
ØØ6A|   MOV 37H,#ØØH
ØØ6D|   MOV DPTR, #ØØ54H
ØØ7Ø|   LCALL ØØ88H
ØØ73|   MOV 36H,82H
ØØ76|   MOV A,36H
ØØ78|   ADD A,#32H
```

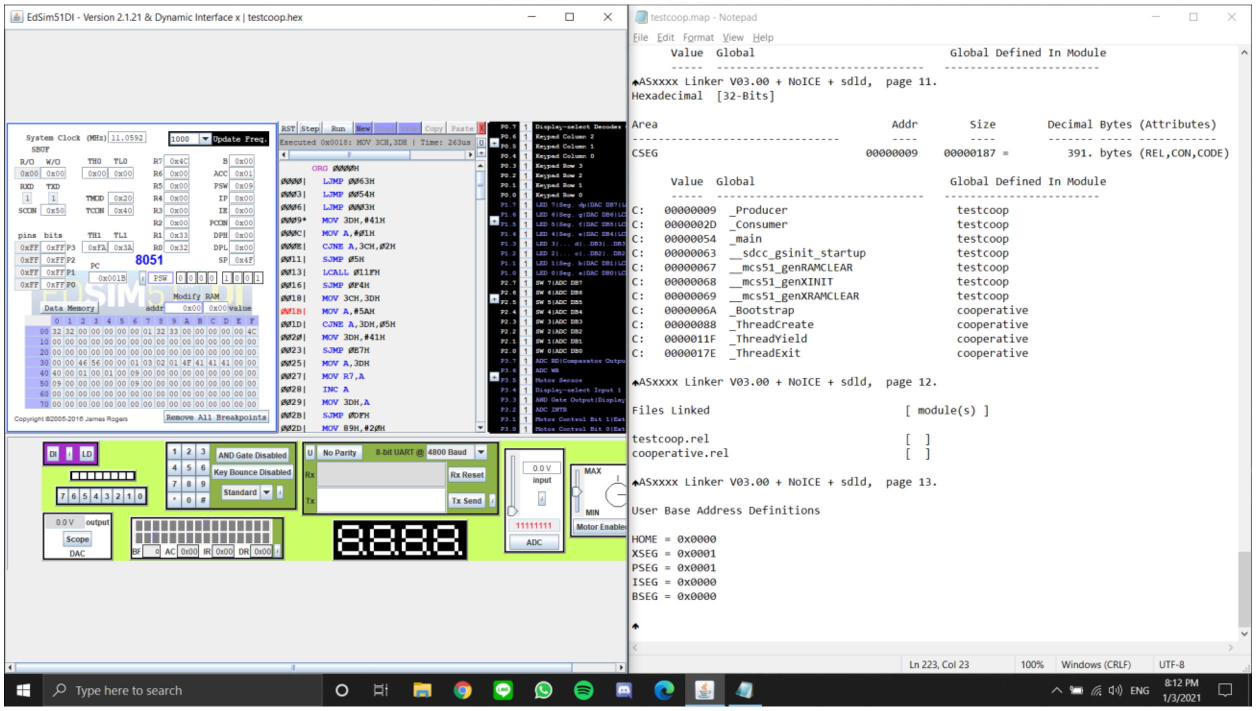
For the stack changes,

when it is running the threadcreate for main, there is no threads created yet. Therefore, thread ID will be set to 0. The address for thread ID 0 is 0x40. Therefore, when pushing DPL, DPH, ACC 4 times and PSW, they all will be pushed to the memory starting at 0x40. This can be seen from the edsim51. After the command reached the line “PUSH 82H” which means PUSH DPL, the data in memory 0x40 that was previously filled with the data 4E turns into 54. When push ACC 4 times is carried out, the data in 0x42 to 0x46 all changed to 00 as ACC is filled with 00. When pushing PSW, the data in 0x46 that was previously filled with 89 now changed to 00.

When it is running the threadcreate for producer, now thread create will take the ID of 1. Thread ID 1 will take the memory starting at 0x50. This means that pushing DPL, DPH, ACC 4 times and PSW will now fill the memory starting 0x50. The rest of the procedures are the same as the previous one.

2. Producer is running

107062381 楊孝偉

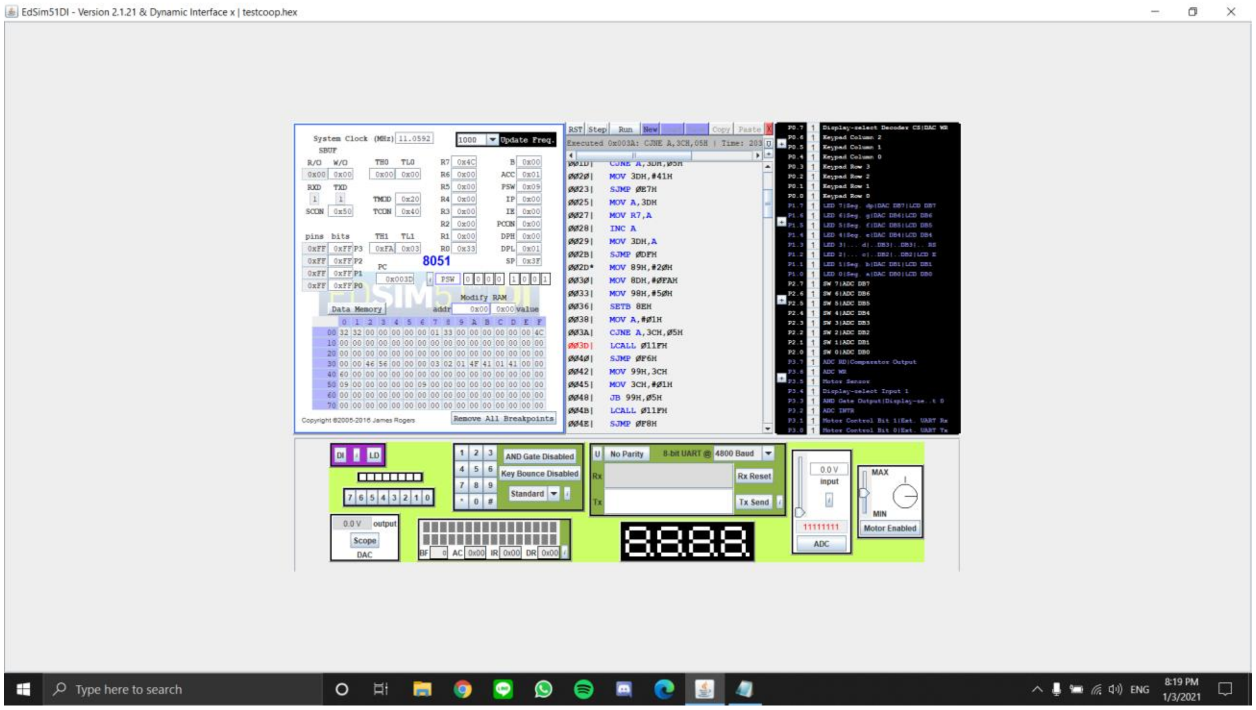


```
Executed 0x0018: MOV 3CH,3DH | Time: 263us
      ORG ØØØØH
ØØØØ|   LJMP ØØ63H
ØØØ3|   LJMP ØØ54H
ØØØ6|   LJMP ØØØ3H
ØØØ9*   MOV 3DH,#41H
ØØØC|   MOV A,#Ø1H
ØØØE|   CJNE A,3CH,Ø2H
ØØ11|   SJMP Ø5H
ØØ13|   LCALL Ø11FH
ØØ16|   SJMP ØF4H
ØØ18|   MOV 3CH,3DH
ØØ1B|   MOV A,#5AH
ØØ1D|   CJNE A,3DH,Ø5H
ØØ2Ø|   MOV 3DH,#41H
ØØ23|   SJMP ØE7H
ØØ25|   MOV A,3DH
ØØ27|   MOV R7,A
ØØ28|   INC A
ØØ29|   MOV 3DH,A
ØØ2B|   SJMP ØDFH
ØØ2D|   MOV 89H,#2ØH
```

```
     Value  Global                              Global Defined In Module
     -----  --------------------------------   ------------------------

ASxxxx Linker V03.00 + NoICE + sdld,  page 11.
Hexadecimal  [32-Bits]

Area                                   Addr       Size        Decimal Bytes (Attributes)
--------------------------------       ----       ----        ------- ----- ------------
CSEG                                   00000009   00000187 =     391. bytes (REL,CON,CODE)

     Value  Global                              Global Defined In Module
     -----  --------------------------------   ------------------------
C:   00000009  _Producer                        testcoop
C:   0000002D  _Consumer                        testcoop
C:   00000054  _main                            testcoop
C:   00000063  __sdcc_gsinit_startup            testcoop
C:   00000067  __mcs51_genRAMCLEAR              testcoop
C:   00000068  __mcs51_genXINIT                 testcoop
C:   00000069  __mcs51_genXRAMCLEAR             testcoop
C:   0000006A  _Bootstrap                       cooperative
C:   00000088  _ThreadCreate                    cooperative
C:   0000011F  _ThreadYield                     cooperative
C:   0000017E  _ThreadExit                      cooperative

ASxxxx Linker V03.00 + NoICE + sdld,  page 12.

Files Linked                              [ module(s) ]

testcoop.rel                              [  ]
cooperative.rel                           [  ]

ASxxxx Linker V03.00 + NoICE + sdld,  page 13.

User Base Address Definitions

HOME = 0x0000
XSEG = 0x0001
PSEG = 0x0001
ISEG = 0x0000
BSEG = 0x0000
```

I know that producer is running since, after the command MOV 3CH 3DH, which means that buffer takes the value in token, the value in 3CH that was previously different from 3DH, now have the same value, which indicates that buffer takes the value in token, which means that producer is running.

3. Consumer is running



```
Executed 0x003A: CJNE A,3CH,05H | Time: 203
ØØ1D|   CJNE A,3DH,Ø5H
ØØ2Ø|   MOV 3DH,#41H
ØØ23|   SJMP ØE7H
ØØ25|   MOV A,3DH
ØØ27|   MOV R7,A
ØØ28|   INC A
ØØ29|   MOV 3DH,A
ØØ2B|   SJMP ØDFH
ØØ2D*   MOV 89H,#2ØH
ØØ3Ø|   MOV 8DH,#ØFAH
ØØ33|   MOV 98H,#5ØH
ØØ36|   SETB 8EH
ØØ38|   MOV A,#Ø1H
ØØ3A|   CJNE A,3CH,Ø5H
ØØ3D|   LCALL Ø11FH
ØØ4Ø|   SJMP ØF6H
ØØ42|   MOV 99H,3CH
ØØ45|   MOV 3CH,#Ø1H
ØØ48|   JB 99H,Ø5H
ØØ4B|   LCALL Ø11FH
ØØ4E|   SJMP ØF8H
```

I know that consumer is running by the command CJNE A 3CH 05H, which means that it checks whether A, that has the value of 1 is the same as 3CH, which is buffer, if yes, then thread yield, which is represented by LCALL 011FH. These commands can only be found in consumer, and since they are carried out by the edsim51, it means that consumer is working.