NIOS-II Instruction Set Simulator

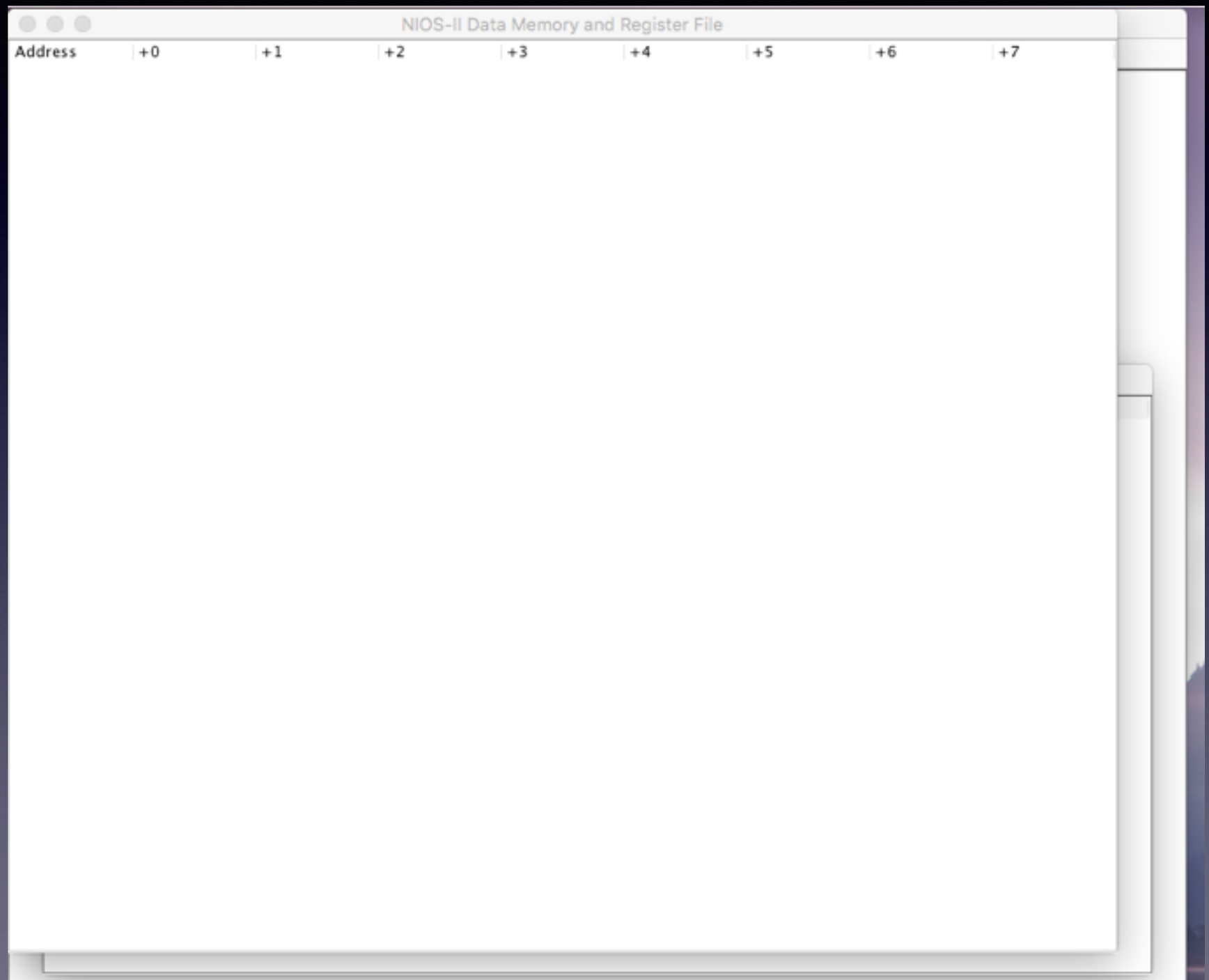
JNISS

Java Archive (JAR)

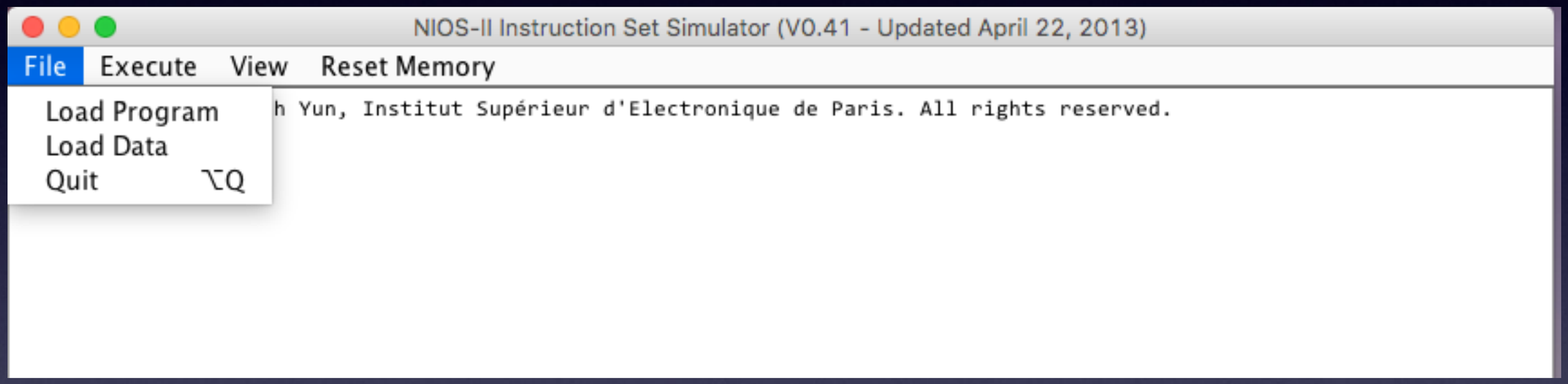
- The instruction set simulator is written in JAVA
- App name : jniss.jar
- To run it from the GUI, need Java Runtime Environment (JRE)
- To run it from the command line, need Java Development Kit (JSK)

jniss.jar

- 3 windows
- Console
- Program Memory
- Registers and Data Memory



Load Program



Program and data must reside in the same file folder as jniss

fact.s

```
        addi   r4, r0, 10
        call   fact              # 10!
        stop

fact:   addi   r27, r27, -32     # allocate stack frame
        stw    r31, 20(r27)
        stw    r28, 16(r27)
        addi   r28, r27, 32

        stw    r4, 0(r28)        # r4 <- n
        blt    r0, r4, L2        # if arg > 0, goto L2
        addi   r2, r0, 1         # fact(0) = 1
        jmpi   L1                # done
L2:     addi   r4, r4, -1        # n--
        call   fact              # fact(n-1)
        ldw    r3, 0(r28)        # r3 <- n
        mul    r2, r2, r3        # fact(n) = n * fact(n-1)

L1:     law    r31, 20(r27)      # restore return address
        ldw    r28, 16(r27)      # restore frame pointer
        addi   r27, r27, 32      # pop stack
        ret                      # jmp ra
```

Console

NIOS-II Instruction Set Simulator (V0.41 - Updated April 22, 2013)

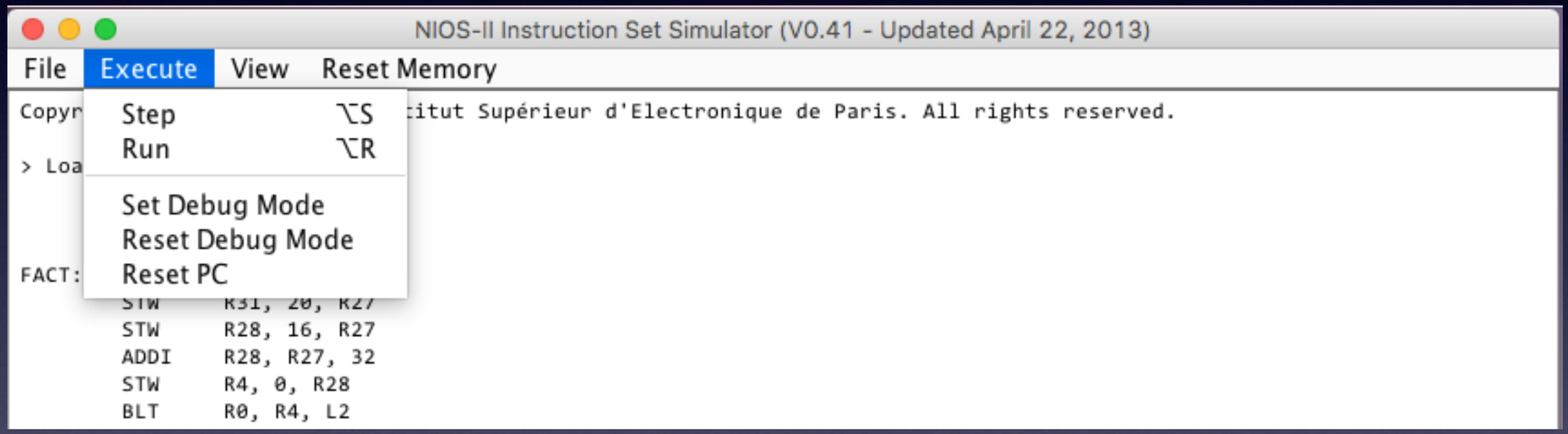
File Execute View Reset Memory

```
Copyright (c) Kenneth Yun, Institut Supérieur d'Electronique de Paris. All rights reserved.

> Load Program fact.s
        ADDI    R4, R0, 10
        CALL    FACT
        STOP
FACT:   ADDI    R27, R27, -32
        STW     R31, 20, R27
        STW     R28, 16, R27
        ADDI    R28, R27, 32
        STW     R4, 0, R28
        BLT     R0, R4, L2
        ADDI    R2, R0, 1
        JMPI    L1
L2:     ADDI    R4, R4, -1
        CALL    FACT
        LDW     R3, 0, R28
        MUL     R2, R2, R3
L1:     LDW     R31, 20, R27
        LDW     R28, 16, R27
        ADDI    R27, R27, 32
        RET

> Display Program Memory
F [000]             4  0 4   ADDI  [rA]  0  [rB]  4  [rC]  0  [offset]        a  [target]            -1
F [001]             0  0 8   CALL  [rA]  0  [rB]  0  [rC]  0  [offset]        0  [target]      FACT   3
F [002]            58  0 1   STOP  [rA]  0  [rB]  0  [rC]  0  [offset]        0  [target]            -1
F [003]      FACT   4  0 4   ADDI  [rA] 27  [rB] 27  [rC]  0  [offset] ffffffe0  [target]            -1
F [004]            21  0 5    STW  [rA] 27  [rB] 31  [rC]  0  [offset]       14  [target]            -1
F [005]            21  0 5    STW  [rA] 27  [rB] 28  [rC]  0  [offset]       10  [target]            -1
F [006]             4  0 4   ADDI  [rA] 27  [rB] 28  [rC]  0  [offset]       20  [target]            -1
F [007]            21  0 5    STW  [rA] 28  [rB]  4  [rC]  0  [offset]        0  [target]            -1
F [008]            22  0 6    BLT  [rA]  0  [rB]  4  [rC]  0  [offset]        0  [target]        L2  11
F [009]             4  0 4   ADDI  [rA]  0  [rB]  2  [rC]  0  [offset]        1  [target]            -1
F [010]             1  0 8   JMPI  [rA]  0  [rB]  0  [rC]  0  [offset]        0  [target]        L1  15
F [011]        L2   4  0 4   ADDI  [rA]  4  [rB]  4  [rC]  0  [offset] ffffffff  [target]            -1
F [012]             0  0 8   CALL  [rA]  0  [rB]  0  [rC]  0  [offset]        0  [target]      FACT   3
F [013]            23  0 5    LDW  [rA] 28  [rB]  3  [rC]  0  [offset]        0  [target]            -1
F [014]            58 39 3    MUL  [rA]  2  [rB]  3  [rC]  2  [offset]        0  [target]            -1
F [015]        L1 23  0 5    LDW  [rA] 27  [rB] 31  [rC]  0  [offset]       14  [target]            -1
F [016]            23  0 5    LDW  [rA] 27  [rB] 28  [rC]  0  [offset]       10  [target]            -1
```

Step/Run


NIOS-II Instruction Set Simulator (V0.41 - Updated April 22, 2013)
File
Execute
View
Reset Memory
Step ⌥S
Run ⌥R
Set Debug Mode
Reset Debug Mode
Reset PC
Copyr
itut Supérieur d'Electronique de Paris. All rights reserved.
> Loa
FACT:
STW R31, 20, R27
STW R28, 16, R27
ADDI R28, R27, 32
STW R4, 0, R28
BLT R0, R4, L2

Program Memory

```
> Step
F [000]          4  0 4   ADDI  [rA]  0  [rB]  4  [rC]  0  [offset]       a  [target]          -1
```

NIOS-II Program Memory

BP	Line	Label	Opcode	rA	rB	rC	Offset	Target
☐	0		ADDI	0	4	0	10	
☐	1		CALL	0	0	0	0	FACT
☐	2		STOP	0	0	0	0	
☐	3	FACT	ADDI	27	27	0	-32	
☐	4		STW	27	31	0	20	
☐	5		STW	27	28	0	16	
☐	6		ADDI	27	28	0	32	
☐	7		STW	28	4	0	0	
☐	8		BLT	0	4	0	0	L2
☐	9		ADDI	0	2	0	1	
☐	10		JMPI	0	0	0	0	L1
☐	11	L2	ADDI	4	4	0	-1	
☐	12		CALL	0	0	0	0	FACT
☐	13		LDW	28	3	0	0	
☐	14		MUL	2	3	2	0	
☐	15	L1	LDW	27	31	0	20	
☐	16		LDW	27	28	0	16	
☐	17		ADDI	27	27	0	32	
☐	18		RET	0	0	0	0	

Set Breakpoint and Run

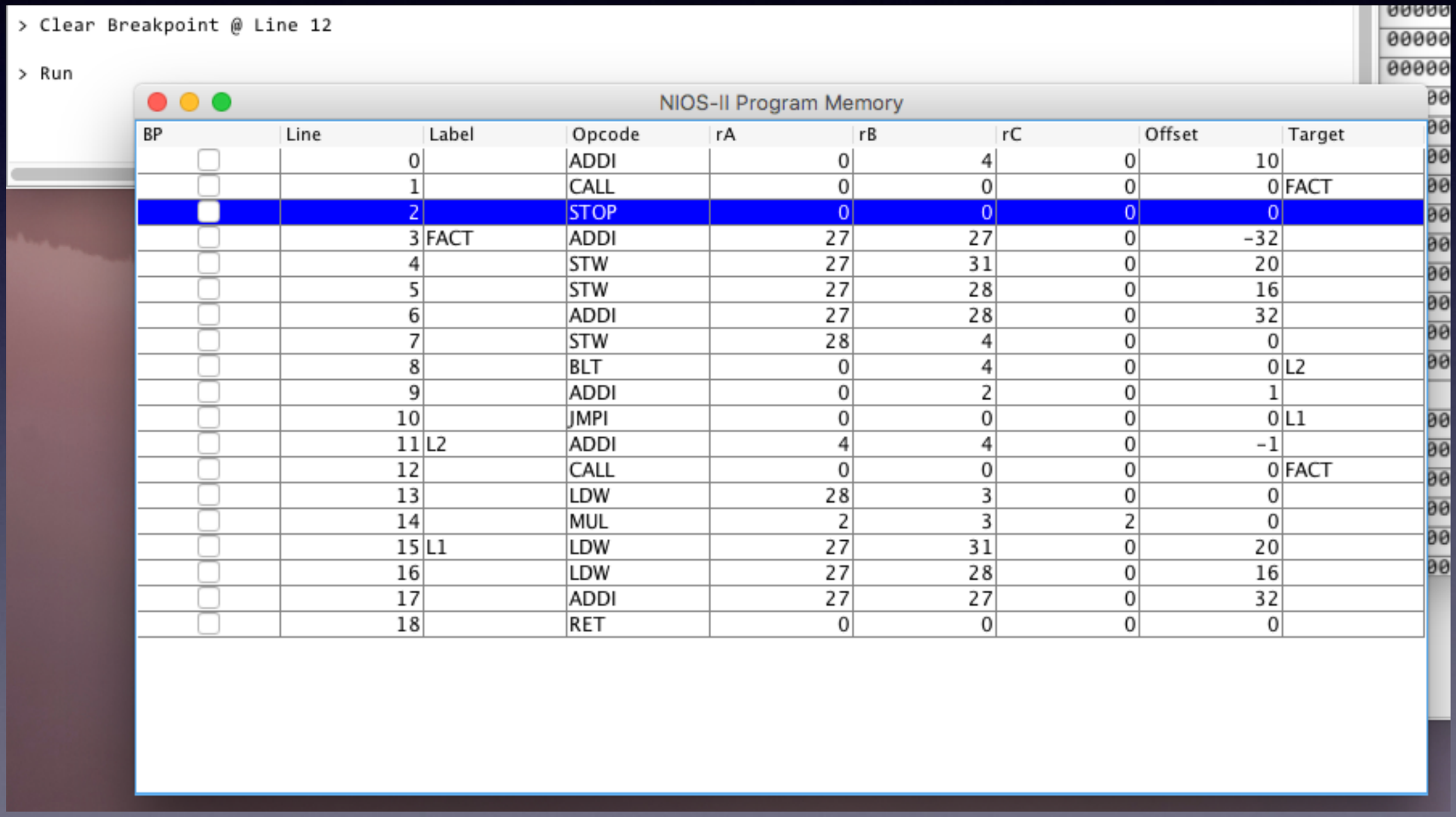
```
> Set Breakpoint @ Line 12

> Run
```

NIOS-II Program Memory

BP	Line	Label	Opcode	rA	rB	rC	Offset	Target
☐	0		ADDI	0	4	0	10	
☐	1		CALL	0	0	0	0	FACT
☐	2		STOP	0	0	0	0	
☐	3	FACT	ADDI	27	27	0	-32	
☐	4		STW	27	31	0	20	
☐	5		STW	27	28	0	16	
☐	6		ADDI	27	28	0	32	
☐	7		STW	28	4	0	0	
☐	8		BLT	0	4	0	0	L2
☐	9		ADDI	0	2	0	1	
☐	10		JMPI	0	0	0	0	L1
☐	11	L2	ADDI	4	4	0	-1	
☑	12		CALL	0	0	0	0	FACT
☐	13		LDW	28	3	0	0	
☐	14		MUL	2	3	2	0	
☐	15	L1	LDW	27	31	0	20	
☐	16		LDW	27	28	0	16	
☐	17		ADDI	27	27	0	32	
☐	18		RET	0	0	0	0	

Clear Breakpoint and Run



BP	Line	Label	Opcode	rA	rB	rC	Offset	Target
☐	0		ADDI	0	4	0	10	
☐	1		CALL	0	0	0	0	FACT
☐	2		STOP	0	0	0	0	
☐	3	FACT	ADDI	27	27	0	-32	
☐	4		STW	27	31	0	20	
☐	5		STW	27	28	0	16	
☐	6		ADDI	27	28	0	32	
☐	7		STW	28	4	0	0	
☐	8		BLT	0	4	0	0	L2
☐	9		ADDI	0	2	0	1	
☐	10		JMPI	0	0	0	0	L1
☐	11	L2	ADDI	4	4	0	-1	
☐	12		CALL	0	0	0	0	FACT
☐	13		LDW	28	3	0	0	
☐	14		MUL	2	3	2	0	
☐	15	L1	LDW	27	31	0	20	
☐	16		LDW	27	28	0	16	
☐	17		ADDI	27	27	0	32	
☐	18		RET	0	0	0	0	

Registers and Data Memory

NIOS-II Data Memory and Register File

Address	+0	+1	+2	+3	+4	+5	+6	+7
Registers								
R[00]	00000000	00000000	00375f00	0000000a	00000000	00000000	00000000	00000000
R[08]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
R[16]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
R[24]	00000000	00000000	0001b000	0001fffc	0001fffc	00000000	00000000	00000002
PC	00000002		GP	SP	FP			RA
Data								
D[0000]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0020]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0040]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0060]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0080]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[00a0]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[00c0]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[00e0]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0100]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0120]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0140]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0160]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[0180]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[01a0]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[01c0]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D[01e0]	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
Stack								
D[1ffe0]	00000000	00000000	00000000	0001fffc	00000002	00000000	00000000	0000000a
D[1ffc0]	00000000	00000000	00000000	0001fffc	0000000d	00000000	00000000	00000009
D[1ffa0]	00000000	00000000	00000000	0001ffdc	0000000d	00000000	00000000	00000008
D[1ff80]	00000000	00000000	00000000	0001ffbc	0000000d	00000000	00000000	00000007
D[1ff60]	00000000	00000000	00000000	0001ff9c	0000000d	00000000	00000000	00000006
D[1ff40]	00000000	00000000	00000000	0001ff7c	0000000d	00000000	00000000	00000005
D[1ff20]	00000000	00000000	00000000	0001ff5c	0000000d	00000000	00000000	00000004