Test Document

Test 1: Testing g, r, and b options through user input

Purpose: This test supplies the program with valid Srecords and setting the breakpoint and running from program counter till the breakpoint address.

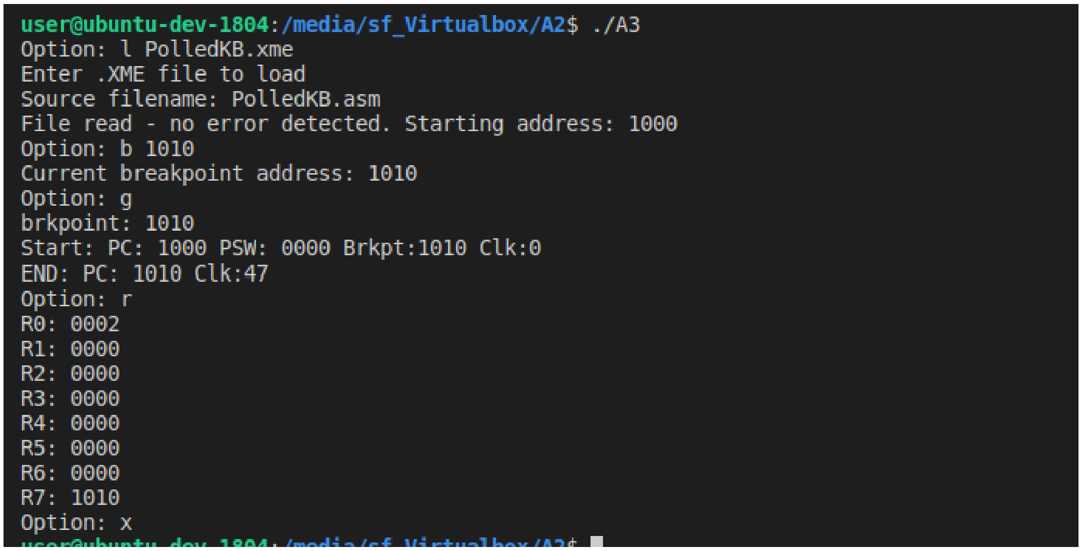
Configuration: The following records are supplied to the program from the **PolledKB.xme** input file:

```
S00F0000506F6C6C65644B422E61736D94
S107000200000000F6
S121100010689260505C826050C102589A470824FC23C180C8C14281DA470824FC2346
S10D101ED26A51454824F223FF234F
S9031000EC
```

Expected results: The expected results are:

```
Option: l PolledKB.xme
Enter .XME file to load
Source filename: PolledKB.asm
File read - no error detected. Starting address: 1000
Option: b 1010
Current breakpoint address: 1010
Option: g
brkpoint: 1010
Start: PC: 1000 PSW: 0000 Brkpt:1010 Clk:0
END: PC: 1010 Clk:47
Option: r
R0: 0002
R1: 0000
R2: 0000
R3: 0000
R4: 0000
R5: 0000
R6: 0000
R7: 1010
Option: x
```

Actual results: The results from the program are:



```
user@ubuntu-dev-1804:/media/sf_Virtualbox/A2$ ./A3
Option: l PolledKB.xme
Enter .XME file to load
Source filename: PolledKB.asm
File read - no error detected. Starting address: 1000
Option: b 1010
Current breakpoint address: 1010
Option: g
brkpoint: 1010
Start: PC: 1000 PSW: 0000 Brkpt:1010 Clk:0
END: PC: 1010 Clk:47
Option: r
R0: 0002
R1: 0000
R2: 0000
R3: 0000
R4: 0000
R5: 0000
R6: 0000
R7: 1010
Option: x
user@ubuntu-dev-1804:/media/sf_Virtualbox/A2$
```

The **test 1 passes**, as it read the input commands g, r and b and stopped the instruction cycle till breakpoint address and display the registers when r is entered.

Test 2: Testing Shift right arthematic (SRA) instruction through Debugger

Test Document

Purpose: This test supplies the program with the command line argument handling i.e reading the options from TEST2.debug and storing various Srecords into the memory and executing the instructions from SRA.xme and also displaying the registers.

Configuration: The following commands are supplied to the program from **TEST2.dbg** input file:

```
l SRA.xme
b 1010
g
r
x
```

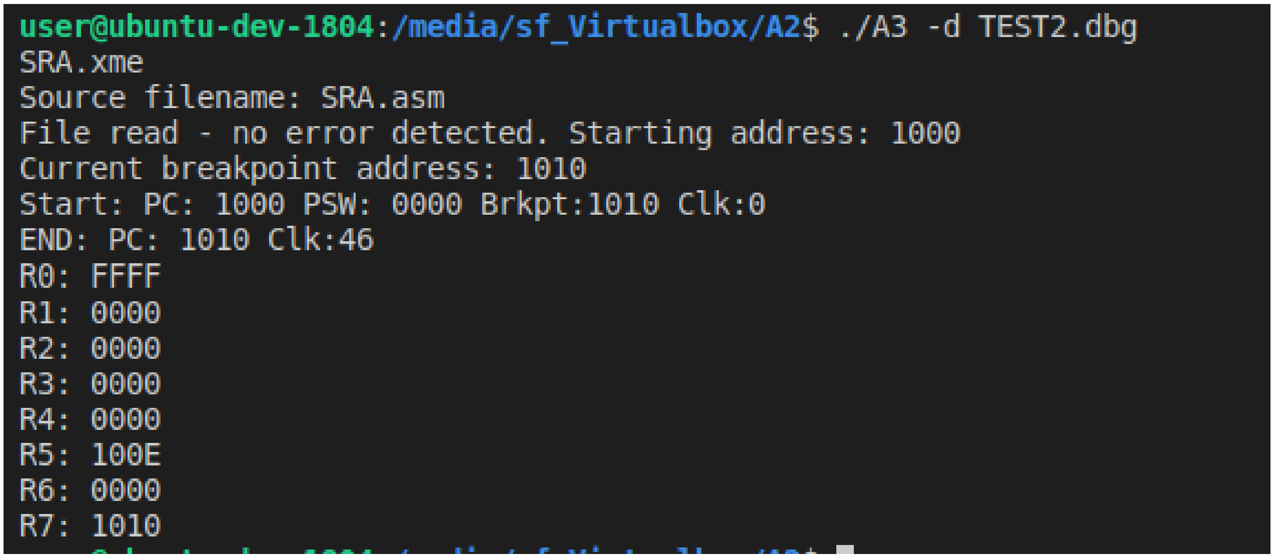
The following records are supplied to the program from **SRA.xme** input file:

```
S00A00005352412E61736DA0
S1050080F0FF8B
S11110008068B8468840004D004D004D004DFC
S9031000EC
```

Expected results: The expected results are:

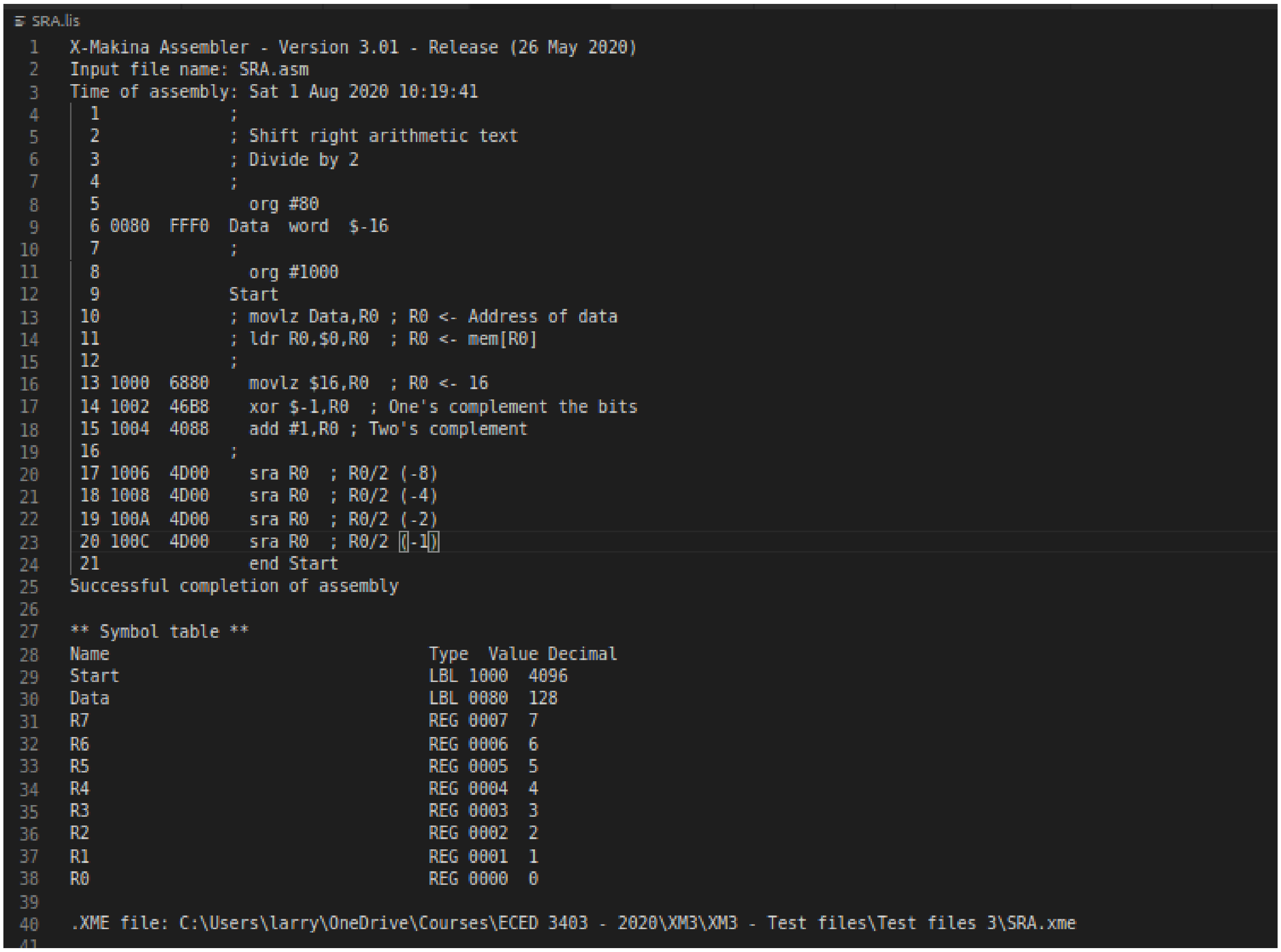
```
SRA.xme
Source filename: SRA.asm
File read - no error detected. Starting address: 1000
Current breakpoint address: 1010
Start: PC: 1000 PSW: 0000 Brkpt:1010 Clk:0
END: PC: 1010 Clk:46
R0: FFFF
R1: 0000
R2: 0000
R3: 0000
R4: 0000
R5: 100E
R6: 0000
R7: 1010
```

Actual results: The results from the program are:



```
user@ubuntu-dev-1804:/media/sf_Virtualbox/A2$ ./A3 -d TEST2.dbg
SRA.xme
Source filename: SRA.asm
File read - no error detected. Starting address: 1000
Current breakpoint address: 1010
Start: PC: 1000 PSW: 0000 Brkpt:1010 Clk:0
END: PC: 1010 Clk:46
R0: FFFF
R1: 0000
R2: 0000
R3: 0000
R4: 0000
R5: 100E
R6: 0000
R7: 1010
```

Now if we compare it to SRA.list file



```
X-Makina Assembler - Version 3.01 - Release (26 May 2020)
Input file name: SRA.asm
Time of assembly: Sat 1 Aug 2020 10:19:41
  1             ;
  2             ; Shift right arithmetic text
  3             ; Divide by 2
  4             ;
  5               org #80
  6 0080  FFF0  Data  word  $-16
  7             ;
  8               org #1000
  9             Start
 10             ; movlz Data,R0 ; R0 <- Address of data
 11             ; ldr R0,$0,R0  ; R0 <- mem[R0]
 12             ;
 13 1000  6880    movlz $16,R0  ; R0 <- 16
 14 1002  46B8    xor $-1,R0  ; One's complement the bits
 15 1004  4088    add #1,R0 ; Two's complement
 16             ;
 17 1006  4D00    sra R0  ; R0/2 (-8)
 18 1008  4D00    sra R0  ; R0/2 (-4)
 19 100A  4D00    sra R0  ; R0/2 (-2)
 20 100C  4D00    sra R0  ; R0/2 (-1)
 21               end Start
Successful completion of assembly

** Symbol table **
Name                                Type  Value Decimal
Start                               LBL 1000  4096
Data                                LBL 0080  128
R7                                  REG 0007  7
R6                                  REG 0006  6
R5                                  REG 0005  5
R4                                  REG 0004  4
R3                                  REG 0003  3
R2                                  REG 0002  2
R1                                  REG 0001  1
R0                                  REG 0000  0

.XME file: C:\Users\larry\OneDrive\Courses\ECED 3403 - 2020\XM3\XM3 - Test files\Test files 3\SRA.xme
```

Here we are storing the value 16 into R0 and then dividing by -8,-4,-2 and then -1 and the expected result is FFFF in R0 which is in actual result as shown in the actual output.
Therefore, **TEST2** passes

Test 3: Testing instructions and displaying it in registers

Test Document

Purpose: This test supplies the program with the Sreccords and then instruction LDR STR are executed and the result is reflected on registers. The instructions movl, movh movlz, STR on LSB, LDR has been tested.

Configuration: The following commands are supplied to the program from **TEST3.dbg** input file:

```
l LDRSTRA3.xme
b 200E
g
r
x
```

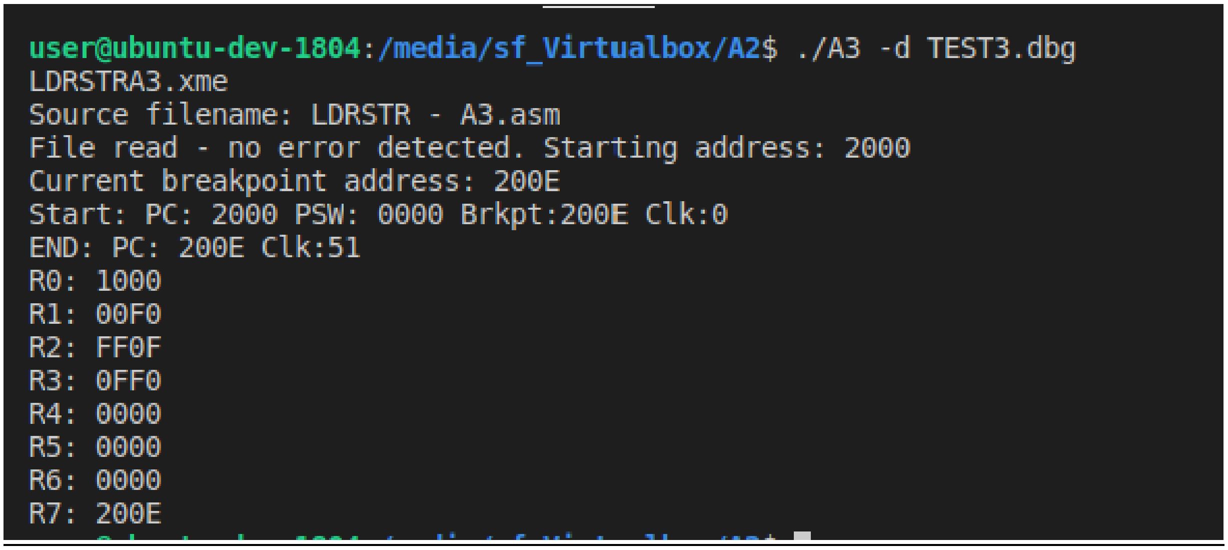
The following Srecords are supplied to the program from **LDRSTRA3.xme** input file:

```
S01200004C4452535452202D2041332E61736DC2
S10510000000EA
S11320000060807881 6F48C07A70D0C00380FF235D
S9032000DC
```

Expected results: The expected results are

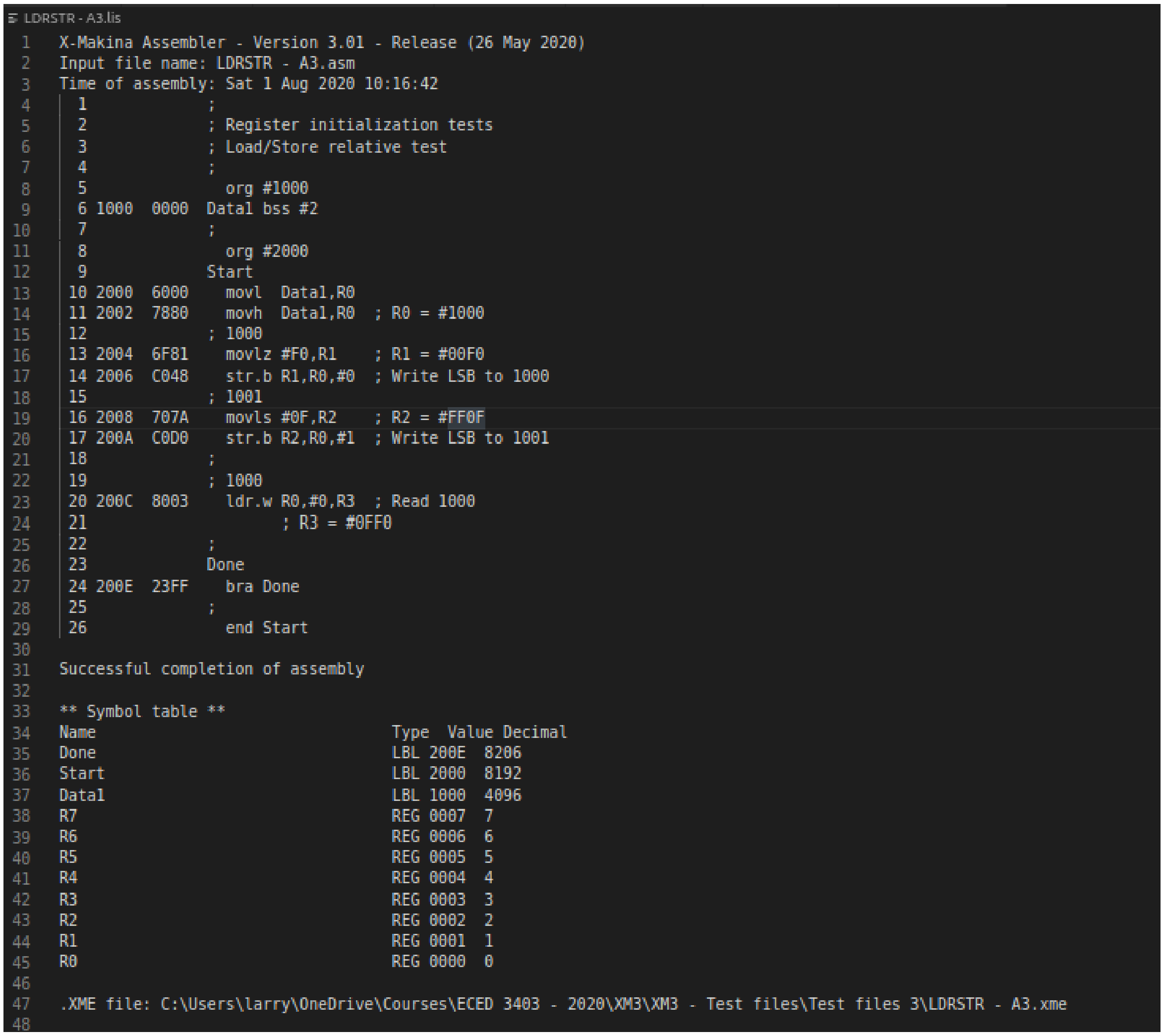
```
LDRSTRA3.xme
Source filename: LDRSTR - A3.asm
File read - no error detected. Starting address: 2000
Current breakpoint address: 200E
Start: PC: 2000 PSW: 0000 Brkpt:200E Clk:0
END: PC: 200E Clk:51
R0: 1000
R1: 00F0
R2: FF0F
R3: 0FF0
R4: 0000
R5: 0000
R6: 0000
R7: 200E
```

Actual results: The results from the program are:



```
user@ubuntu-dev-1804:/media/sf_Virtualbox/A2$ ./A3 -d TEST3.dbg
LDRSTRA3.xme
Source filename: LDRSTR - A3.asm
File read - no error detected. Starting address: 2000
Current breakpoint address: 200E
Start: PC: 2000 PSW: 0000 Brkpt:200E Clk:0
END: PC: 200E Clk:51
R0: 1000
R1: 00F0
R2: FF0F
R3: 0FF0
R4: 0000
R5: 0000
R6: 0000
R7: 200E
```

Now if we look at the **LDRSTR - A3.list** file,



```
X-Makina Assembler - Version 3.01 - Release (26 May 2020)
Input file name: LDRSTR - A3.asm
Time of assembly: Sat 1 Aug 2020 10:16:42
  1               ;
  2               ; Register initialization tests
  3               ; Load/Store relative test
  4               ;
  5                 org #1000
  6 1000  0000  Data1 bss #2
  7               ;
  8                 org #2000
  9             Start
 10 2000  6000    movl  Data1,R0
 11 2002  7880    movh  Data1,R0  ; R0 = #1000
 12               ; 1000
 13 2004  6F81    movlz #F0,R1    ; R1 = #00F0
 14 2006  C048    str.b R1,R0,#0  ; Write LSB to 1000
 15               ; 1001
 16 2008  707A    movls #0F,R2    ; R2 = #FF0F
 17 200A  C0D0    str.b R2,R0,#1  ; Write LSB to 1001
 18               ;
 19               ; 1000
 20 200C  8003    ldr.w R0,#0,R3  ; Read 1000
 21                     ; R3 = #0FF0
 22               ;
 23             Done
 24 200E  23FF    bra Done
 25               ;
 26                 end Start

Successful completion of assembly

** Symbol table **
Name                                  Type  Value Decimal
Done                                  LBL 200E  8206
Start                                 LBL 2000  8192
Data1                                 LBL 1000  4096
R7                                    REG 0007  7
R6                                    REG 0006  6
R5                                    REG 0005  5
R4                                    REG 0004  4
R3                                    REG 0003  3
R2                                    REG 0002  2
R1                                    REG 0001  1
R0                                    REG 0000  0

.XME file: C:\Users\larry\OneDrive\Courses\ECED 3403 - 2020\XM3\XM3 - Test files\Test files 3\LDRSTR - A3.xme
```

From line number 10 till 21 R0,R1,R2 and R3 value has been changed so if we look at the registers from the actual result we will find same value i.e:

```
R0: 1000
R1: 00F0
R2: FF0F
R3: 0FF0
R4: 0000
R5: 0000
R6: 0000
R7: 200E (Program counter)
```

Hence, **test 3** passes.

<u>TEST4: Control C catcher</u>

Test Document

Purpose: This test supplies the program with Srecords but the breakpoint address will be far bigger then the starting address therefore in order to stop the program control-C is entered.

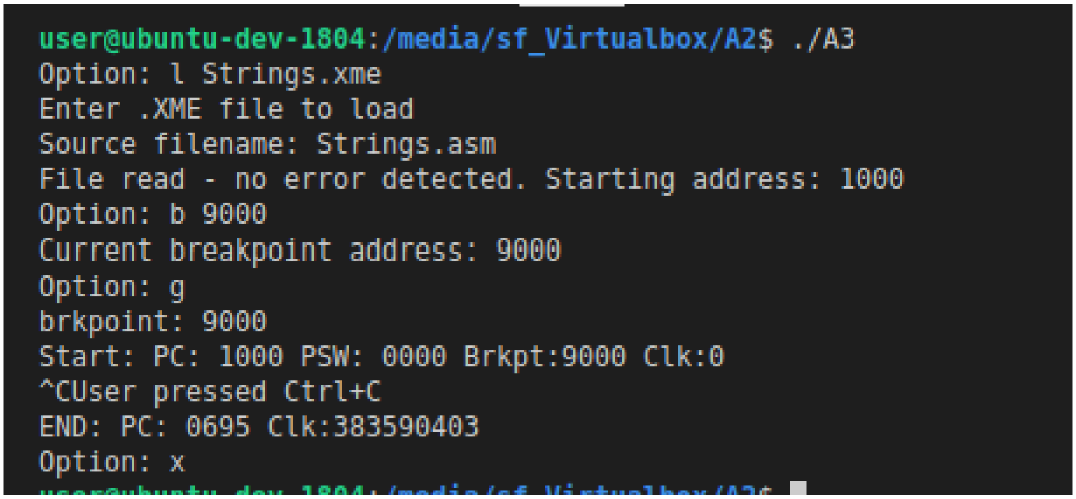
The following Srecords are supplied to the program from **Strings.xme** input file:

```
S00E0000537472696E67732E61736D98
S121100006761160817907009142815C01FF52608279F607FF230E5F165F0068CA58F8
S111101EC24550248840FB23B258B1582F4CD1
S11F2000065F0E5F165F03688B40C858C25CF8474824FA23B258B158B0582F4CA7
S10E300000005468652063617400004 8
S9031000EC
```

Expected results: The expected results are:

```
Option: l Strings.xme
Enter .XME file to load
Source filename: Strings.asm
File read - no error detected. Starting address: 1000
Option: b 9000
Current breakpoint address: 9000
Option: g
brkpoint: 9000
Start: PC: 1000 PSW: 0000 Brkpt:9000 Clk:0
^CUser pressed Ctrl+C
END: PC: 0695 Clk:383590403
Option: x
```

Actual results: The results from the program are:



```
user@ubuntu-dev-1804:/media/sf_Virtualbox/A2$ ./A3
Option: l Strings.xme
Enter .XME file to load
Source filename: Strings.asm
File read - no error detected. Starting address: 1000
Option: b 9000
Current breakpoint address: 9000
Option: g
brkpoint: 9000
Start: PC: 1000 PSW: 0000 Brkpt:9000 Clk:0
^CUser pressed Ctrl+C
END: PC: 0695 Clk:383590403
Option: x
user@ubuntu-dev-1804:/media/sf_Virtualbox/A2$
```

The **test4 passes** as we that the program stopped when ctrl-C was entered.