

# Linking Multiple mic1 Source Files

```
-bash-4.1$ cat main.asm
main:  lodd arg1:
       push
       call lcshft:
       stod rslt:
       halt
       .LOC 8
arg1:  7
rslt:  0
```

```
-bash-4.1$ cat lcshft.asm
lcshft: lodl 1
        jneg add1:
        addl 1
        retn
add1:  addl 1
        add c1:
        retn
c1:    1
```

lcshft will do a shift of its argument (on stack) **one bit circular left**, with the high displaced bit looped to the low bit position in the result. The result is returned in the AC.

e.g. **0**000 0000 0000 0111 → 0000 0000 0000 111**0**

e.g. **1**001 1100 1110 0000 → 0011 1001 1100 000**1**

```

-bash-4.1$ cat main.asm
main:    lodd arg1:
         push
         call lcshft:
         stod rslt:
         halt
         .LOC 8
arg1:    7
rslt:    0

```

```

-bash-4.1$ cat lcshft.asm
lcshft: lodl 1
         jneg add1:
         addl 1
         retn
add1:   addl 1
         add  c1:
         retn
c1:     1

```

```

-$ ./masm_mrd -o < main.asm
0  U00000000000000000000 arg1:
1  111101000000000000
2  U11100000000000000000 lcshft:
3  U00010000000000000000 rslt:
4  111111111110000000
8  0000000000000000111
9  0000000000000000000
4096 x
rslt:    9
arg1:    8
main:    0

```

```

-$ ./masm_mrd -o < lcshft.asm
0  1000000000000000001
1  U11000000000000000000 add1:
2  1010000000000000001
3  11111000000000000000
4  1010000000000000001
5  U00100000000000000000 c1:
6  11111000000000000000
7  0000000000000000001
4096 x
c1:     7
add1:   4
lcshft: 0

```

```
-bash-4.1$ ./linker -s main.obj lcshift.obj
```

FIRST PASS

```
 0  U000000000000000000  arg1:      main:      lodd arg1:
 1  11110100000000000000                                push
 2  U111000000000000000  lcshft:   call  lcshft:
 3  U000100000000000000  rslt:     stod  rslt:
 4  111111111110000000                                halt  .LOC 8
 8  000000000000000111                                arg1:     7
 9  000000000000000000                                rslt:     0
-----
10  100000000000000001                                lcshft:   lodl  1
11  U110000000000000000  add1:     jneg  add1:
12  101000000000000001                                addl  1
13  111110000000000000                                retn
14  101000000000000001                                add1:     addl  1
15  U001000000000000000  c1:       addd  c1:
16  111110000000000000                                retn
17  000000000000000001                                c1:       1
```

RELOCATED SYMBOL TABLE

Relative Symbols		Relative Symbols	
main.asm		Lcshift.asm	
rslt:	9	c1:	7
arg1:	8	add1:	4
main:	0	lcshft:	0



add1:	14
arg1:	8
c1:	17
lcshft:	10
main:	0
rslt:	9

## FIRST PASS

```
0  U000000000000000000  arg1:
1  111101000000000000
2  U111000000000000000  lcshft:
3  U000100000000000000  rslt:
4  11111111111000000
```

```
8  00000000000000111
9  000000000000000000
10 100000000000000001
11 U110000000000000000  add1:
12 101000000000000001
13 111110000000000000
14 101000000000000001
15 U001000000000000000  c1:
16 111110000000000000
17 000000000000000001
```

## SECOND PASS

```
0: 0000000000001000 =8
1: 111101000000000000
2: 11100000000001010 =10
3: 0001000000001001 =9
4: 111111111110000000
5: 111111111111111111
6: 111111111111111111
7: 111111111111111111
8: 00000000000000111
9: 000000000000000000
10: 100000000000000001
11: 1100000000001110 =14
12: 101000000000000001
13: 111110000000000000
14: 101000000000000001
15: 00100000000010001 =17
16: 111110000000000000
17: 000000000000000001
```

### RELOCATED SYMBOLS

```
add1: 14
arg1: 8
c1: 17
lcshft: 10
main: 0
rslt: 9
```

```
-bash-4.1$ ./masm_mrd -o < main.asm > main.obj
-bash-4.1$ ./masm_mrd -o < lcshift.asm > lcshift.obj
-bash-4.1$ ./linker main.obj lcshift.obj > my_prog.exe
-bash-4.1$ cat my_prog.exe
```

```
00000000000001000
1111010000000000
11100000000001010
00010000000001001
1111111111000000
11111111111111111
11111111111111111
11111111111111111
00000000000000111
00000000000000000
10000000000000001
11000000000001110
10100000000000001
11111000000000000
10100000000000001
00100000000010001
11111000000000000
00000000000000001
```

```
-bash-4.1$ ./mic1 prom_mrd_v1.dat my_prog.exe 0 1024
```

```
Read in 152 micro instructions
```

```
Read in 18 machine instructions
```

```
Starting PC is : 0000000000000000 base 10: 0
```

```
Starting SP is : 0000010000000000 base 10: 1024
```

```
ProgramCounter : 0000000000000101 base 10: 5
```

```
Accumulator : 0000000000001110 base 10: 14
```

```
InstructionReg : 1111111111000000 base 10: 65472
```

```
TempInstr : 1000000000000000 base 10: 32768
```

```
StackPointer : 0000001111111111 base 10: 1023
```

```
ARegister : 1010001111111111 base 10: 41983
```

```
BRegister : 0000000000000000 base 10: 0
```

```
CRegister : 0000000000000000 base 10: 0
```

```
DRegister : 0000000000000000 base 10: 0
```

```
ERegister : 0000000000000000 base 10: 0
```

```
FRegister : 0000000000000000 base 10: 0
```

```
Total cycles : 91
```

```
Type decimal address, q to quit or c to continue: 9
```

```
the location 9 has value 0000000000001110 , or 14 or signed 14
```