**nullProcess Program Assembly Language w/ Comments**

Label Mnemonic Operands Description

Main Function Start of main function

Loop Branch Loop Branch to Loop

End Loop Execution starts at Loop Function

**nullProcess Program Symbol Table**

**Written by: Jonathon Ku**

|  |  |
| --- | --- |
| **Label** | **Address** |
| main | 0 |
| Loop | 0 |

**nullProcess Program Machine Code w Comments**

**Written by: Jonathon Ku**

Address Content Comment

1 66000 Loop; Branch to Loop

2 1 Address of Loop

3 0 Halt

-1 1 End of Program; PC = 1

**Program 1 Assembly Language with Comments**

**Written by: Jonathon Ku**

Label Mnemonic Operands Description

main Function Start of main function

R Long 4 Declare variable R and initialize it 4

M Long 3 Declare variable M and initialize it 3

Count Long 150 Declare variable Count and initialize it 149

Start Move R1++, R R1 = R, R1 Autoincrement mode

Move R2, 150 R2 = 150. Used for MemAllocSystemCall

System Call 4 System Call using ID = 4, MemAllocSystemCall

Loop Add R, M R = R + M

Multiply R, -1 R = R\*-1

Move R1++, R R1 = R, R1 Autoincrement mode

Subtract Count, 1 Count = Count -1

BrOnPlus Count, Loop If Count > 0 jump to Loop

System Call 5 System Call using ID = 5, MemFreeSystemCall

Halt Halt the program

End Start Execution starts at Move Instruction

**Program 1 Symbol Table**

**Written by: Jonathon Ku**

|  |  |
| --- | --- |
| **Label** | **Address** |
| main | 0 |
| R | 0 |
| M | 1 |
| Count | 2 |
| Loop | 9 |

**Program 1 Machine Code w/ Comments**

**Written by: Jonathon Ku**

Address Content Comment

0 4 R long 4; variable R set to 4, address starts at default 0

1 3 M long 3; variable M set to 3, address 1

2 150 Count long 150; variable Count set to 150, address 2

3 53150 Start; Move R1++,R; set R1 to R, increment R1

4 0 Address of R

5 51260 Move R2, 150; set R2 to 150

6 150 Immediate Operand 150

7 126000 System Call

8 4 System Call ID = 4, Memory Allocation

9 15050 Loop; Add R, M; R = R + M

10 0 Address of R

11 1 Address of M

12 35060 Multiply R, -1; R = R \* -1

13 0 Address of R

14 -1 Immediate Operand -1

15 53150 Move R1++, R; set R1 to R, increment R1

16 0 Address of R

17 25060 Subtract Count, 1; Count = Count - 1

18 2 Address of Count

19 1 Immediate Operand 1

20 85000 Branch on Plus Count, Loop

21 2 Address of Count

22 9 Address of Loop

23 126000 System Call

24 5 System Call ID = 5, Memory Free

25 0 Halt

-1 3 End of Program, PC = 3

**Program 2 Assembly Language with Comments**

**Written by: India Ervin**

**Revised by: Jonathon Ku**

Label Mnemonic Operand Description

main Function Start of main function

Count Long 8 Declare variable Count and initialize it to 8

Start Move R2,10 R2 = 10

Loop Push 1094 Push 1094 onto Stack  
Push 4206 Push 1094 onto Stack  
Push 9197 Push 1094 onto Stack  
Push 2813 Push 1094 onto Stack  
Push 3724 Push 1094 onto Stack  
Push 5003 Push 1094 onto Stack  
Push 1444 Push 1094 onto Stack  
Push 4444 Push 1094 onto Stack  
Push 4448 Push 1094 onto Stack  
Push 2404 Push 1094 onto Stack  
Pop R3 Pop from Stack, store in R3  
Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Pop R3 Pop from Stack, store in R3

Subtract Count, 1 Count = Count - 1  
BrOnPlus Count, Loop If Count > 0, jump to Loop  
Halt Halt the program

End Start Execution starts at Move instruction.

**Program 2 Symbol Table**

**Written by: India Ervin**

**Revised by: Jonathon Ku**

|  |  |
| --- | --- |
| **Label** | **Address** |
| main | 0 |
| Count | 0 |
| Loop | 3 |

**Program 2 Machine Code w/ Comments**

**Written by: India Ervin**

**Revised by: Jonathon Ku**

Address Content Comment   
0 8 Count long 8; Set variable count to 8   
1 51260 Move R2, 10; Set R2 to 10  
2 10 Immediate operand value of 10   
3 106000 Loop; Push 1094  
4 1094 Immediate operand value of 1094  
5 106000 Push 4206  
6 4206 Immediate operand value of 4206  
7 106000 Push 9197  
8 9197 Immediate operand value of 9197  
9 106000 Push 2813  
10 2813 Immediate operand value of 2813  
11 106000 Push 3724  
12 3724 Immediate operand value of 3724  
13 106000 Push 5003  
14 5003 Immediate operand value of 5003  
15 106000 Push 1444  
16 1444 Immediate operand value of 1444  
17 106000 Push 4444  
18 4444 Immediate operand value of 4444  
19 106000 Push 4448  
20 4448 Immediate operand value of 4448  
21 106000 Push 2404  
22 2404 Immediate operand value of 2404  
23 111300 Pop, store in R3  
24 111300 Pop, store in R3  
25 111300 Pop, store in R3  
26 111300 Pop, store in R3  
27 111300 Pop, store inR3  
28 111300 Pop, store in R3  
29 111300 Pop, store in R3  
30 111300 Pop, store in R3  
31 111300 Pop, store in R3  
32 111300 Pop, store in R3  
33 25060 Subtract Count by 1  
34 0 Address of Count  
35 1 Immediate operand value of 1  
36 85000 If count > 0

37 3 Address of Loop  
38 0 Halt  
-1 1 End of Program PC = 1