**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 | 4 |
| R | 4 |
| M | 5 |
| Count | 6 |
| Loop | 13 |

**Program 1 Machine Code w/ Comments**

**Written by: Jonathon Ku**

Address Content Comment

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

5 3 M long 3; variable M set to 3, address 1

6 150 Count long 150; variable Count set to 150, address 2

7 51260 Move R2, 150; set R2 to 150

8 150 Immediate Operand 150

9 126000 System Call

10 4 System Call ID = 4, Memory Allocation

11 53150 Start; Move R1++,R; set R1 to R, increment R1

12 4 Address of R

13 15050 Loop; Add R, M; R = R + M

14 4 Address of R

15 5 Address of M

16 35060 Multiply R, -1; R = R \* -1

17 4 Address of R

18 -1 Immediate Operand -1

19 53150 Move R1++, R; set R1 to R, increment R1

20 4 Address of R

21 25060 Subtract Count, 1; Count = Count - 1

22 6 Address of Count

23 1 Immediate Operand 1

24 85000 Branch on Plus Count, Loop

25 6 Address of Count

26 13 Address of Loop

27 126000 System Call

28 5 System Call ID = 5, Memory Free

29 0 Halt

-1 7 End of Program, PC = 7

**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 | 30 |
| Count | 30 |
| Loop | 33 |

**Program 2 Machine Code w/ Comments**

**Written by: India Ervin**

**Revised by: Jonathon Ku**

Address Content Comment   
30 8 Count long 8; Set variable count to 8   
31 51260 Move R2, 10; Set R2 to 10  
32 10 Immediate operand value of 10   
33 106000 Loop; Push 1094  
34 1094 Immediate operand value of 1094  
35 106000 Push 4206  
36 4206 Immediate operand value of 4206  
37 106000 Push 9197  
38 9197 Immediate operand value of 9197  
39 106000 Push 2813  
40 2813 Immediate operand value of 2813  
41 106000 Push 3724  
42 3724 Immediate operand value of 3724  
43 106000 Push 5003  
44 5003 Immediate operand value of 5003  
45 106000 Push 1444  
46 1444 Immediate operand value of 1444  
47 106000 Push 4444  
48 4444 Immediate operand value of 4444  
49 106000 Push 4448  
50 4448 Immediate operand value of 4448  
51 106000 Push 2404  
52 2404 Immediate operand value of 2404  
53 111300 Pop, store in R3  
54 111300 Pop, store in R3  
55 111300 Pop, store in R3  
56 111300 Pop, store in R3  
57 111300 Pop, store inR3  
58 111300 Pop, store in R3  
59 111300 Pop, store in R3  
60 111300 Pop, store in R3  
61 111300 Pop, store in R3  
62 111300 Pop, store in R3  
63 25060 Subtract Count by 1  
64 33 Address of Count  
65 1 Immediate operand value of 1  
66 85000 If count > 0

67 33 Address of Loop  
68 0 Halt  
-1 31 End of Program PC = 31

**Program 3 Assembly Language w/ Comments**

**Written by: Gabe Freitas**

**Revised by: Jonathon Ku**

Label Mnemonic Operands Description

main Function Start of main function

Count Long 5 Declare variable Count and initialize it 549

Start Move R2, 9 R2 = 9. Used for MemAllocSystemCall

System Call 4 System Call using ID = 4, MemAllocSystemCall

Move R3, R1 R3 = R1

Input System Call 8 System Call using ID = 8, Input Interrupt

Move R3++, R0 R3 = R0, R3 Autoincrement mode

Subtract Count, 1 Count = Count - 1

BrOnPlus Count, Input If Count > 0, jump to Input

Move Count, 5 Count = 5

Move R3, R1 R3 = R1

Move R0, R3++ R0 = R3, R3 Autoincrement mode

Output System Call 9 System Call using ID = 9, Output Interrupt

Subtract Count, 1 Count = Count - 1

BrOnPlus Count, Output If Count > 0, jump to Output

System Call 5 System Call using ID = 5, MemFreeSystemCall

Halt Halt the program

End Start Execution begins at the Move instruction

**Program 3 Symbol Table**

**Written by: Gabe Freitas**

**Revised by: Jonathon Ku**

|  |  |
| --- | --- |
| **Label** | **Address** |
| main | 69 |
| Count | 69 |
| Start | 70 |
| Input | 75 |
| Output | 88 |

**Program 3 Machine Code w/ Comments**

**Written by: Gabe Freitas**

**Revised by: Jonathon ku**

Address Content Comment

69 5 Long Count = 5

70 51260 Start; Move R2, 9; R2 = 9

71 9 Immediate Operand 9

72 126000 System Call

73 4 System Call ID 4: Mem Allocation

74 51311 Move R3, R1; R3 = R1

75 126000 Input; System Call

76 8 System Call ID 8: Input Interrupt

77 53310 R3++ = R0

78 25060 Subtract Count, 1; Count = Count - 1

79 69 Address of Count

80 1 Immediate Operand 1

81 85000 Count > 0, Branch to Input

82 69 Address of Count

83 75 Address of Input

84 55060 Move Count, 5; Count = 5

85 69 Address of Count

86 5 Immediate Operand 5

87 51311 Move R3, R1; R3 = R1

88 51033 Output; Move R0, R3++; R0 = R3++

89 126000 System Call

90 9 System Call ID 9: Output Interrupt

91 25060 Subtract Count, 1; Count = Count -1

92 69 Address of Count

93 1 Immediate Operand 1

94 85000 Count > 0, Branch to Output

95 69 Address of Count

96 88 Address of Output

97 126000 System Call

98 5 System Call ID 5: Mem Free

99 0 Halt

-1 70 End of Program, PC = 70