Eddy Olu, Matthew Joseph, and Vito Parisi

Below are the Assembly program, Symbol table, and Machine code with comments in that order

Null program was done by Matthew Joseph

Program #1 was done by Matthew Joseph

Program #2 was done by Vito Parisi

Program #3 was done by Eddy Olu

Null program written by Matthew Joseph

| Label | Mnemonic | operands | description |
| --- | --- | --- | --- |
| main | Function |  | Start of main function |
|  | Origin | 0 |  |
| LoopStart | Branch | 1 | Branch to LoopStart |
|  | End Of Program | Start | The program execution starts at the start |

Symbol table for Null program

| Symbol | Value (Address) |
| --- | --- |
| main | 0 |
| LoopStart | 1 |

Machine code for Null program with comments

1 66000 //loop start

2 1 //loop location

-1 0 //EOP The program execution starts at 0

Program #1 assembly written by Matthew Joseph

| Label | Mnemonic | operands | description |
| --- | --- | --- | --- |
| main | Function |  | Start of main function |
|  | Origin | 3 | state where the start of the program is. |
| Start | Move | GPR2, 100 | Move GPR2,100; move the constant 100 into GPR 2. |
|  | System call | 4 | system call memory allocate |
|  | Branch on zero | GPR0, BranchLoc1 | If the value of GPR0 is zero, then jump to address of BranchLoc1 |
|  | Branch | EndOfProgram | jump to EndOfProgram |
| BranchLoc1 | Move | GPR7,GPR1 | Move GPR7,GPR1; move value from register 1 to register 7 |
|  | Move | GPR3,3 | Move GPR3,3; move the constant 3 into register 3 |
|  | Move | GPR4,-1 | move the constant -1 into register 4 |
|  | Move | Pointer,GPR3 | move value of GPR 3 into the memory location in GPR7 then increment GPR7 by 1 |
|  | Subtract | GPR2,1 | subtract constant 1 from register 2, used to count spaces |
| BranchLoc2 | Multiply | GPR3,GPR4 | multiply reg3 and reg4 |
|  | Move | GPR7,GPR3 | move value of GPR 3 into the memory location in GPR7 and then increment GPR7 by 1 |
|  | Subtract | GPR2,1 | subtract 1 from reg2 |
|  | Branch on plus | GPR2,BranchLoc2 | if value in reg2 is positive jump BranchLoc2 |
|  | Move | GPR2,100 | Move 100 into gpr 2 |
|  | System call | 5 | System call to free spaces in gpr 2 which is 100 |
|  | Halt |  | CPU will halt |
|  | End Of Program | Start | The program execution starts at the start |

Symbol table for Program #1

| Symbol | Value (Address) |
| --- | --- |
| main | 3 |
| BranchLoc1 | 11 |
| BranchLoc2 | 19 |

Program #1 machine code with comments

3 51260 // Move GPR2,100; move the constant 100 into GPR 2.

4 100 // Immediate value of op2

5 126000 //system call function

6 4 //system call memory allocate

7 91000 //BrOnZero BranchLoc1; If the value of GPR0 is zero, then jump to

8 11 //Address of BranchLoc1

9 60000 //Branch; jump to EndOfProgram

10 29 //Address of EndOfProgram

11 51711 //Move GPR7,GPR1; move value from register 1 to register 7

12 51360 //Move GPR3,3; move the constant 3 into register 3

13 3 //Immediate value of op2

14 51460 //Move GPR4,-1; move the constant -1 into register 4

15 -1 //Immediate value of op2

16 53713 //move value of GPR 3 into the memory location in GPR7 and then increment GPR7 by 1

17 21260 //Sub GPR2,1; subtract constant 1 from register 2, used to count spaces

18 1 //decrement reg2 by 1

19 31314 //multiply reg3 and reg4, BranchLoc2

20 53713 //move value of GPR 3 into the memory location in GPR7 and then increment GPR7 by 1

21 21260 //subtract 1 from reg2

22 1

23 81250 // BrOnPlus if value in reg2 is positive

24 19 // mem location to branch to

25 51260 //move next value into register 2

26 100

27 126000 //System call

28 5 //System call 5 free memory

29 0 //EndOfProgram

-1 3 //The program execution starts at 3

------------------------------------------------------------------------------------------------------------------------

Program #2 assembly Written by Vito Paris

| Label | Mnemonic | operands | description |
| --- | --- | --- | --- |
| main | function |  | Start of main function |
|  | Origin | 30 | state where the start of the program is. |
| Start | Move | GPR1,8 | Move 8 into register 1 |
| Loop | Push | 7001 | push 7001 onto stack |
|  | Push | 7002 | push 7002 onto stack |
|  | Push | 7003 | push 7003 onto stack |
|  | Push | 7004 | push 7004 onto stack |
|  | Pop |  | pop value on top of stack |
|  | Pop |  | pop value on top of stack |
|  | Pop |  | pop value on top of stack |
|  | Pop |  | pop value on top of stack |
| LoopCounter | Subtract | GPR1,1 | subtract next value from reg 1 |
|  | Branch on plus | GPR1 | If reg 1 is positive branch to Loop |
|  | Halt |  | CPU will halt |
|  | End Of Program | Start | The program execution starts at the start |

Symbol table for Program #2

| Symbol | Value (Address) |
| --- | --- |
| main | 30 |
| Loop | 32 |
| LoopCounter | 48 |

Program #2 machine code with comments

30 51160 //move next line into register 1

31 8 //move 8 into register 1, LoopCounter

32 106000 //push next value onto stack, Loop

33 7001 //1234 pushed onto stack

34 106000 //push next value onto stack

35 7002 //1235 pushed onto stack

36 106000 //push next value onto stack

37 7003 //1236 pushed onto stack

38 106000 //push next value onto stack

39 7004 //1237 pushed onto stack

40 115000 //pop value on top of stack

41 0

42 115000 //pop value on top of stack

43 0

44 115000 //pop value on top of stack

45 0

46 115000 //pop value on top of stack

47 0

48 21160 //subtract next value from reg 1, LoopCounter

49 1

50 81150 //BrOnPlus check LoopCounter

51 32 //Branch to Loop

52 0 //EOP

-1 30 // The program execution starts at 30

---------------------------------------------------------------------------------------------------------------------------

Program #3 assembly Written by Eddy Olu

| Label | Mnemonic | operands | description |
| --- | --- | --- | --- |
| main | function |  | Start of main function |
|  | Origin | 53 | state where the start of the program is. |
| Start | Move | GPR2,10 | move 10 into register 2 |
|  | Move | GPR5,5 | move next value to reg5 LoopCheck |
|  | Systemcall | 4 | MemAlloc systemcall |
|  | Move | GPR1,GPR6 | move contents of reg1 to reg6 |
|  | Move | GPR1,GPR7 | move contents of reg1 to reg7 |
| LoopStart | SystemCall |  | SystemCall IO\_GetC |
|  | Move | GPR7,GPR1 | move contents of reg 1 into reg 7 and also auto increment 7 |
| LoopCheck | Subtract | GPR5 | subtract 1 from reg 5 |
|  | Branch on plus | GPR5 | Check if reg 5 is positive branch to LoopStart |
|  | Move | GPR5,5 | move 5 to reg5 |
| Loop2 | Move | GPR1,GPR6 | move reg1 to reg6 and increment |
|  | System Call | 9 | SysCall IO\_PutC |
|  | Subtract | GPR5,1 | subtract from 1 from reg 5 LoopCheck |
|  | Branch on plus | GPR5 | If reg 5 is positive branch to Loop2 |
|  | Subtract | GPR6,5 | Subtract 5 from trg 6 |
|  | Move | GPR1,GPR6 | move value of reg 1 to reg6 |
|  | System call | 5 | SysCall Mem\_Free |
|  | Halt |  | CPU will halt |
|  | End Of Program | Start | The program execution starts at the start |

Symbol table for Program #3

| Symbol | Value (Address) |
| --- | --- |
| main | 53 |
| LoopStart | 61 |
| Loop2 | 70 |

Program #3 machine code with comments

53 51260 //move next value into register 2

54 10 //move 10 to reg2, allocate 10 spaces in memory

55 51560 //move next value to reg5

56 5 //move 5 to reg5, LoopCheck

57 126000 //systemcall

58 4 //syscall 4 MemAlloc

59 51611 //move contents of reg1 to reg6

60 51711 //move contents of reg1 to reg7

61 126000 //SystemCall IO\_GetC, LoopStart

62 8 //SysCall IO\_GetC

63 53711 //move contents of reg 1 into reg 7 and also auto increment 7

64 21560 //subtract next line from reg 5, LoopCheck

65 1 //loopDec

66 81550 // BrOnPlus check LoopCheck

67 61 //Branch to LoopStart

68 51560 //move next line to reg5

69 5

70 51136 //move reg1 to reg6 and increment, Loop2

71 126000 //System Call

72 9 //SysCall IO\_PutC

73 21560 //subtract from LoopCheck

74 1 //loopDec

75 81550 //BranchonPlus checks LoopCheck

76 70 //branch to Loop2

77 21660 //decrement next line from reg6

78 5 //loopender

79 51116 //move value of reg 1 to reg6

80 126000 //system call

81 5 //SysCall Mem\_Free

82 0 //EOP

-1 53// The program execution starts at 53

----------------------------------------------------------------------------------------------------------