**Brendan Burdick**

**Operating Systems Internals (Homework 2)**

**Professor Suban Krishnamoorthy**

**Assembly Code With Comments, HYPO Executable with Comments, and Symbol Table**

**NULL PROGRAM:**

**ASSEMBLY CODE WITH COMMENTS:**

**Label Mnemonic Operands Description**

Null Function Start of null function.

Origin 0 Start null function at location 0.

Loop Branch Loop Branch to Loop label.

Halt Halt the program execution.

End Loop End of program, PC = 0.

**SYMBOL TABLE:**

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| Null | 0 |
| Loop | 0 |

**HYPO EXECUTABLE WITH COMMENTS:**

**Address Content Comment**

0 60000 Address of Loop label. Branch instruction found.

1 0 Branch to address zero (Loop).

2 0 Halt.

-1 0 PC = 0.

**PROGRAM 1- Dynamic Memory Allocation and Freeing:**

**ASSEMBLY CODE WITH COMMENTS:**

**Label Mnemonic Operands Description**

Main Function Start of main function.

Origin 3 Start null function at location 3.

Move GPR2, 100 Move 100 into GPR2.

SystemCall 4 Call **mem-alloc** syscall (attempt to allocate 100 locations).

BrOnZero GPR0, AddressOk If GPR0 = 0, successful allocation, branch.

Branch EndProg Otherwise, branch to the end of the program.

AddressOk Move GPR7, GPR1 Move GPR1 into GPR7.

Move GPR3, 3 Move 3 into GPR3 (This is ‘N’).

Move GPR4, 5 Move 5 into GPR4 (This is ‘M’).

Move (GPR7)++, GPR3 Store GPR3 value into address in GPR7, auto-increment.

Subtract GPR2, 1 Subtract GPR2 by 1.

StartOfLoop Add GPR3, GPR4 Add GPR3 and GPR5 (GPR3 = GPR3 + GPR5).

Move (GPR7)++, GPR3 Store GPR3 value into address in GPR7, auto-increment.

Subtract GPR2, 1 Subtract GPR2 by 1.

BrOnPlus GPR2, StartOfLoop If GPR2 > 0, loop.

Move GPR2, 100 Move 100 into GPR2.

SystemCall 5 Free 100 memory locations.

EndProg Halt Halt the program execution.

End Main Program ends.

**SYMBOL TABLE:**

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| Main | 3 |
| AddressOk | 11 |
| StartOfLoop | 19 |
| EndProg | 29 |

**HYPO EXECUTABLE WITH COMMENTS:**

**Address Content Comment**

3 51260 Move 100 into GPR2 (where PC should start).

4 100 Immediate value 100.

5 126000 Systemcall command.

6 4 Mem-alloc system call. Attempt to allocate 100 locations.

7 91000 BrOnZero, if GPR0 = 0, allocation was successful, branch.

8 11 Branch to location 11.

9 60000 Otherwise, if GPR0 != 0, branch.

10 29 Branch to location 29, end of program.

11 51711 Move GPR1 value into GPR7 (so GPR7 has start address).

12 51360 Move 3 into GPR3 (This is ‘N’).

13 3 Immediate value 3.

14 51460 Move 5 into GPR4. (This is ‘M’).

15 5 Immediate value 5.

16 53713 Move GPR3 value into memory address within GPR7, auto-increment.

17 21260 Subtract GPR2 by 1 (GPR2 = GPR2 – 1).

18 1 Immediate value 1.

19 11314 Add GPR3 and GPR4 (GPR3 = GPR3 + GPR4).

20 53713 Move GPR3 value into memory address within GPR7, auto-increment.

21 21260 Subtract GPR2 by 1 (GPR2 = GPR2 – 1).

22 1 Immediate value 1.

23 81250 BrOnPlus, if GPR2 > 0, branch.

24 19 Branch to location 19 (address of ‘StartOfLoop’).

25 51260 Move 100 into GPR2.

26 100 Immediate value 100.

27 126000 Systemcall command.

28 5 Mem-free system call. Attempt to free 100 locations.

29 0 Halt.

-1 3 PC = 3.

**PROGRAM 2- Pushing and Popping Stack Values:**

**ASSEMBLY CODE WITH COMMENTS:**

**Label Mnemonic Operands Description**

Main Function Start of main function.

Origin 30 Start null function at location 3.

Move GPR1, 35 Move 35 into GPR1 (This is ‘N’).

LoopStart Push 1234 Push 1234 onto the stack.

Push 1235 Push 1235 onto the stack.

Push 1236 Push 1236 onto the stack.

Push 1237 Push 1237 onto the stack.

Pop 2700 Pop into arbitrary location 2700.

Pop 2700 Pop into arbitrary location 2700.

Pop 2700 Pop into arbitrary location 2700.

Pop 2700 Pop into arbitrary location 2700.

Subtract GPR1, 1 Subtract GPR1 by 1 (GPR1 = GPR1 – 1).

BrOnPlus GPR1, LoopStart If GPR1 > 0, loop.

Halt Halt the program execution.

End Main Program ends.

**SYMBOL TABLE:**

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| Main | 30 |
| LoopStart | 32 |

**HYPO EXECUTABLE WITH COMMENTS:**

**Address Content Comment**

30 51160 Move 35 into GPR1.

31 35 Immediate value of 35 (This is ‘N’).

32 106000 Push 1234 onto the stack.

33 1234 Immediate value of 1234

34 106000 Push 1235 onto the stack.

35 1235 Immediate value of 1235

36 106000 Push 1236 onto the stack.

37 1236 Immediate value of 1236

38 106000 Push 1237 onto the stack.

39 1237 Immediate value of 1237

40 115000 Pop value from top of stack (will be 1237).

41 2700 Store into arbitrary memory location 2700.

42 115000 Pop value from top of stack (will be 1236).

43 2700 Store into arbitrary memory location 2700.

44 115000 Pop value from top of stack (will be 1235).

45 2700 Store into arbitrary memory location 2700.

46 115000 Pop value from top of stack (will be 1234).

47 2700 Store into arbitrary memory location 2700.

48 21160 Subtract 1 from GPR1.

49 1 Immediate value of 1.

50 81150 BrOnPlus, if GPR1 > 0, branch (address of ‘LoopStart’).

51 32 Branch to location 32, and loop.

52 0 Halt.

-1 30 PC = 30.

**PROGRAM 3- Performing Input / Output Operations:**

**ASSEMBLY CODE WITH COMMENTS:**

**Label Mnemonic Operands Description**

Main Function Start of main function.

Origin 53 Start null function at location 53.

Move GPR2, 10 Move 10 into GPR2 (for memory allocation).

Move GPR5, 5 Move 5 into GPR5 (for 5-letter word).

iStart SystemCall 4 Call **mem-alloc** syscall (attempt to allocate 10 locations).

Move GPR6, GPR1 Move GPR1 into GPR6.

Move GPR7, GPR1 Move GPR1 into GPR7.

SystemCall 8 Call io\_getc system call.

Move (GPR7)++, GPR1 Move GPR1 value into address in GPR7, auto-increment.

Subtract GPR5, 1 Subtract GPR5 by 1.

BrOnPlus GPR5, iStart If GPR5 > 0, branch to iStart.

Move GPR5, 5 Move 5 into GPR5.

oStart Move GPR1, (GPR6)++ Move content of address in GPR6 to GPR1, auto-increment

SystemCall 9 Call io\_putc system call.

Subtract GPR5, 1 Subtract GPR5 by 1.

BrOnPlus GPR5, oStart If GPR5 > 0, branch to oStart.

Subtract GPR6, 5 Subtract GPR6 by 5.

Move GPR1, GPR6 Move GPR6 into GPR1.

SystemCall 5 Free 100 memory locations.

Halt Halt the program execution.

End Main Program ends.

**SYMBOL TABLE:**

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| Main | 53 |
| iStart | 61 |
| oStart | 70 |

**HYPO EXECUTABLE WITH COMMENTS:**

**Address Content Comment**

53 51260 Move 10 into GPR2.

54 10 Immediate value 10.

55 51560 Move 5 into GPR5.

56 5 Immediate value 5.

57 126000 Systemcall command.

58 4 Call **mem-alloc** syscall (attempt to allocate 10 locations).

59 51611 Move GPR1 into GPR6.

60 51711 Move GPR1 into GPR7.

61 126000 Systemcall command.

62 8 Call io\_getc systcall. Attempt to read a character.

63 53711 Move GPR1 value into the address in GPR7. Auto-increment.

64 21560 Subtract GPR5 by 1 (GPR5 = GPR5 – 1).

65 1 Immediate value 1.

66 81550 BrOnPlus, if GPR5 > 0, branch.

67 61 Branch to address 61 (address of ‘iStart’).

68 51560 Move 5 into GPR5.

69 5 Immediate value 5.

70 51136 Move the value of the address in GPR6 into GPR1. Auto-increment.

71 126000 Systemcall command.

72 9 Call io\_putc systcall. Attempt to print a character.

73 21560 Subtract GPR5 by 1 (GPR5 = GPR5 – 1).

74 1 Immediate value 1.

75 81550 BrOnPlus, if GPR5 > 0, branch.

76 70 Branch to address 61 (address of ‘oStart’).

77 21660 Subtract GPR6 by 5 (GPR6 = GPR6 – 5).

78 5 Immediate value 5.

79 51116 Move GPR6 into GPR1.

80 126000 Systemcall command.

81 5 Free 10 memory locations.

82 0 Halt.

-1 53 PC = 53.