# **CS 410 Binary to Assembly Activity Template**

**Ismael DeRocco**

**Southern New Hampshire Univercity**

**CS410**

**7/16/2023**

**Step 1:** List the binary file name.

**Step 2:** Identify the functions in the binary file.

**Step 3**: Convert the binary file to assembly code.

**Step 4:** Align the blocks of assembly code with their corresponding function in the binary file.

**Step 5:** Explain the functionality of the blocks of assembly code.

## File One: [assignment3\_1.o]

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main() | push %rbp | Pushes a value of %rbp to stack |
| mov %rsp,%rbp | Move %rsp value to %rbp |
| mov $0x400634,%edi | Move first string to %edi |
| callq 0x400450 <puts@plt> | Calls the first string to be printed |
| mov $0x400648,%edi | Move second string to %edi |
| callq 0x400450 <puts@plt> | Calls the second String to be printed |
| mov $0x40065c,%edi | Moves third string to %edi |
| callq 0x400450 <puts@plt> | Calls the second string to be printed. |
| mov $0x0,%edi | Moves 0x0 to %edi |
| callq 0x400480 <exit@plt> | Exit Program |

## File Two: [assignment3\_2.o]

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main() | push %rbp | Push value %rbp to stack |
| mov %rsp,%rbp | Move %rsp to %rbp |
| sub $0x20,%rsp | Subtract 20 from %rsp |
| mov %fs:0x28,%rax | Move 28 to %rax |
| mov %rax,-0x8(%rbp) | Move %rax to -0x0(%rbp) |
| xor %eax,%eax | Assign 0 of %eax to %eax |
| mov $0x400714,%edi | Move first string to %edi |
| callq 0x4004e0 <puts@plt> | Calls first string |
| lea -0x20(%rbp),%rax | Puts -0x20(%rbp) to %rax |
| mov %rax,%rsi | Moves value of %rax to %rsi |
| mov $0x40072b,%edi | Moves second string to %edi |
| mov $0x0,%eax | Moves 0 to %eax |
| callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf to read a string |
| lea -0x20(%rbp),%rax | Puts -0x20(%rbp) to %rax |
| mov %rax,%rsi | Move %rax to %rsi |
| mov $0x40072e,%edi | Moves third string to %edi |
| mov $0x0,%eax | Moves 0 to %eax |
| callq 0x4004f0 <printf@plt> | Calll a print string |
| mov $0x0,%edi | Moves 0 to %edi |
| callq 0x400530 <exit@plt> | Exit program |

## File Three: [assignment3\_3.o]

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| AddNumbers() | push %rbp | Push value of %rbp to stack |
| mov %rsp,%rbp | Move %rsp to %rbp |
| mov %edi,-0x4(%rbp) | Move %edi to -0x4(%rbp) |
| mov %esi,-0x8(%rbp) | Move %esi to -0x8(%rbp) |
| mov -0x8(%rbp),%eax | Move -0x8(%rbp) to %eax |
| mov -0x4(%rbp),%edx | Move -0x4(%rbp) to %edx |
| add %edx,%eax | Add values of %edx and %eax |
| pop %rbp | Restore main’s copy from stack |
| retn | Retun the value of %rbp |
| Main() | push %rbp | Pushes value %rbp to stack |
| mov %rsp,%rbp | Moves %rsp to %rbp |
| sub $0x10,%rsp | Subtracts 10 from %rsp |
| mov $0x400734,%edi | Moves string to %edi |
| callq 0x4004e0 <puts@plt> | Calls to print the string |
| lea -0x8(%rbp),%rdx | Place -0x8(%rbp) in %rdx |
| lea -0xc(%rbp),%rax | Place -0xc(%rbp) in %rax |
| mov %rax,%rsi | Moves %rax value into %rsi |
| mov $0x400747,%edi | Moves second string into %edi |
| mov $0x0,%eax | Moves 0 to %rax |
| callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf to read string |
| mov -0x8(%rbp),%edx | Place -0x8(%rbp) in %edx |
| mov -0xc(%rbp),%eax | Place -0xc(%rbp) in %eax |
| mov %edx,%esi | Moves %edx to %esi |
| mov %eax,%edi | Moves %eax to %edi |
| callq 0x40062d <AddNumbers> | Calls addNumber Function |
| mov %eax,-0x4(%rbp) | Moves %eax to -0x4(%rbp) |
| mov -0x8(%rbp),%edx | Moves -0x8(%rbp) into %edx. |
| mov -0xc(%rbp),%eax | Moves -0xc(%rbp) into %eax. |
| mov -0x4(%rbp),%ecx | Moves -0x4(%rbp) into %ecx. |
| mov %eax,%esi | Moves %eax to %esi |
| mov $0x40074d,%edi | Moves $0x40074d to %edi |
| mov $0x0,%eax | Moves 0 to %eax |
| callq 0x4004f0 <printf@plt> | Calls printed string |
| mov $0x0,%edi | Moves 0 to %edi |
| callq 0x400530 <exit@plt> | Exit program |

## File Four: [assignment3\_4.o]

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| PrintFact() | push %rbp | Pushes value of %rbp to stack. |
| mov %rsp,%rbp | Moves value of %rsp to %rbp. |
| sub $0x20,%rsp | Subtracts 20 from %rsp. |
| mov %edi,-0x14(%rbp) | Moves %edi to -0x14(%rbp). |
| movl $0x1,-0x4(%rbp) | Moves 1 into -0x4(%rbp). |
| mov -0x14(%rbp),%eax | Moves -0x14(%rbp) into %eax. |
| mov %eax,-0x8(%rbp) | Moves %eax into -0x8(%rbp). |
| jmp 0x400669 <PrintFact+60> | Jump to 0x400669 (line 60). |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| imul -0x8(%rbp),%eax | Multiplies -0x8(%rbp) by %eax. |
| mov %eax,-0x4(%rbp) | Moves %eax into -0x4(%rbp). |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| mov %eax,%esi | Moves %eax into %esi. |
| mov $0x400844,%edi | Moves the string into %edi. |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x4004f0 <printf@plt> | Calls print string. |
| subl $0x1,-0x8(%rbp) | Subtracts 1 from -0x8(%rbp). |
| cmpl $0x0,-0x8(%rbp) | Compares 0 and the value of  -0x8(%rbp). |
| jg 0x400647 <PrintFact+26> | Jump to 0x400657, line 26 if greater then 0. |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| mov %eax,%esi | Moves %eax into %esi. |
| mov $0x400848,%edi | Moves string into %edi. |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x4004f0 <printf@plt> | Calls print string. |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| leaveq | Copies frame pointer to the stack pointer and releases the stacked space |
| retq | Restores the value of %rip to the value saved on the stack. |
| PrintSum() | push %rbp | Pushes value of %rbp to stack. |
| mov %rsp,%rbp | Moves %rsp into %rbp. |
| sub $0x20,%rsp | Subtracts 20 from %rsp. |
| mov %edi,-0x14(%rbp) | Moves %edi into -0x14(%rbp). |
| movl $0x0,-0x4(%rbp) | Moves 0 into -0x4(%rbp). |
| mov -0x14(%rbp),%eax | Moves -0x14(%rbp) into %eax. |
| mov %eax,-0x8(%rbp) | Moves %eax into -0x8(%rbp). |
| jmp 0x4006c0 <PrintSum+56> | Jumps to 0x4006c0 (Line 56). |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| add %eax,-0x4(%rbp) | Adds %eax and -0x4(%rbp). |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| mov %eax,%esi | Moves %eax into %esi. |
| mov $0x400844,%edi | Moves string into %edi. |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x4004f0 <printf@plt> | Calls print string. |
| subl $0x1,-0x8(%rbp) | Subtracts 1 from -0x8(%rbp). |
| cmpl $0x0,-0x8(%rbp) | Compares 0 and the value of -0x8(%rbp) |
| jg 0x4006a2 <PrintSum+26> | Jump to 0x4006a2, line 26 if greater then 0. |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| mov %eax,%esi | Moves %eax into %esi. |
| mov $0x400848,%edi | Moves string into %edi. |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x4004f0 <printf@plt> | Calls print string. |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| leaveq | Copies frame pointer to the stacker pointer and releases the stack space. |
| retq | Retsores the vaule of %rip to the value saved on the stack. |
| DisplayMenu() | push %rbp | Pushes value of %rbp to stack. |
| mov %rsp,%rbp | Moves %rsp into %rbp. |
| mov $0x400851,%edi | Moves first string to %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| mov $0x400864,%edi | Moves second string to %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| mov $0x400871,%edi | Moves third string to %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| mov $0x40087e,%edi | Moves fourth string to %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| mov $0x400851,%edi | Moves fifth string to %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| pop %rbp | Restore main’s copy from the stack. |
| retq | Restores the value of the %rip to the value saved on the stack. |
| Main() | push %rbp | Pushes value of %rbp to stack. |
| mov %rsp,%rbp | Moves %rsp into %rbp. |
| sub $0x10,%rsp | Subtracts 10 from %rsp. |
| movl $0x0,-0x8(%rbp) | Moves 0 into -0x8(%rbp). |
| jmp 0x4007a0 <main+137> | Jumps to 0x4007a0 (Line 137). |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x4006df <DisplayMenu> | Calls the DisplayMenu function. |
| mov $0x400886,%edi | Moves first string into %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| lea -0x8(%rbp),%rax | Places -0x8(%rbp) into %rax. |
| mov %rax,%rsi | Moves %rax into %rsi. |
| mov $0x400899,%edi | Moves second string into %edi. |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf to read a string. |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| cmp $0x3,%eax | Compares 3 with value of %eax. |
| je 0x40077a <main+99> | Jump to 0x40077a (Line 99) if cmp is equl |
| mov $0x40089c,%edi | Move third string to %edi. |
| callq 0x4004e0 <puts@plt> | Calls print string. |
| lea -0x4(%rbp),%rax | Places -0x4(%rbp) into %rax. |
| mov %rax,%rsi | Moves %rax into %rsi. |
| mov $0x400899,%edi | Moves fourth string into %edi. |
| mov $0x0,%eax | Moves 0 into %eax. |
| callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf to read a string. |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| cmp $0x1,%eax | Compares 1 with value of %eax. |
| jne 0x40078e <main+119> | Jumps to 0x40078e (Line 119) if cmp is not equal |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| mov %eax,%edi | Moves %eax into %edi. |
| callq 0x40062d <PrintFact> | Calls PrintFact function. |
| jmp 0x4007a0 <main+137> | Jumps to 0x4007a0 (Line 137). |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| cmp $0x2,%eax | Compares 2 with value of %eax. |
| jne 0x4007a0 <main+137> | Jumps to 0x4007a0 (Line 137) if cmp is not equal. |
| mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax. |
| mov %eax,%edi | Moves %eax into %edi. |
| callq 0x400688 <PrintSum> | Calls PrintSum function. |
| mov -0x8(%rbp),%eax | Moves -0x8(%rbp) into %eax. |
| cmp $0x3,%eax | Compares 3 with value of %eax. |
| jne 0x400728 <main+17> | Jumps to 0x400728 (Line 17) if cmp if cmp is not equal. |
| mov $0x0,%edi | Move 0 to %edi |
| callq 0x400530 <exit@plt> | Exit Program |