CS 410 Project One Proficiency Test Template

Explain the functionality of the blocks of assembly code.

“main” function”

Assembly Code Block	Explanation of Functionality
push %rbp mov %rsp,%rbp lea 0x5eb(%rip),%rsi # 0x555555401440 lea 0x201244(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_E S5_PKc@plt>	Sets up the stack frame and outputs “Hello! Welcome to our investment company”
callq 0x555555400f6a <_Z25CheckUserPermissionAccessv> mov %eax,0x201494(%rip) # 0x555555602300 <answer>	Calls boolean function CheckUserPermissionAccess() Assigns return value to “access”
mov 0x20148e(%rip),%eax # 0x555555602300 <answer> cmp $0x1,%eax je 0x555555400e8a <main+64>	Compares value of “access” to 1(true) If values are equal, execution jumps to main+64
lea 0x5f2(%rip),%rsi # 0x555555401470 lea 0x20121b(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_E S5_PKc@plt>	If previous comparison is not equal to 1(true) outputs “Permission Denied.”
lea 0x5ff(%rip),%rsi # 0x5555554014b0 lea 0x2011e8(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_E S5_PKc@plt> lea 0x614(%rip),%rsi # 0x5555554014d8 lea 0x2011d5(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_E S5_PKc@plt> lea 0x625(%rip),%rsi # 0x5555554014fc lea 0x2011c2(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_E S5_PKc@plt>	Outputs the menu of the application

lea 0x20140a(%rip),%rsi # 0x5555556022f4 <choice> lea 0x2012cf(%rip),%rdi # 0x5555556021c0 <_ZSt3cin@@GLIBCXX_3.4> callq 0x555555400c60 <_ZNSirsERi@plt> lea 0x61d(%rip),%rsi # 0x55555540151a lea 0x20119c(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> mov %rax,%rdx mov 0x2013e2(%rip),%eax # 0x5555556022f4 <choice> mov %eax,%esi mov %rdx,%rdi callq 0x555555400d00 <_ZNSolsEi@plt> mov %rax,%rdx mov 0x2010aa(%rip),%rax # 0x555555601fd0 mov %rax,%rsi mov %rdx,%rdi callq 0x555555400ca0 <_ZNSolsEPFRSoS_E@plt>	Takes in user input and saves to variable “userChoice” Outputs “You chose ‘userChoice’” with a newline
mov 0x2013bd(%rip),%eax # 0x5555556022f4 <choice> cmp $0x1,%eax jne 0x555555400f43 <main+249> callq 0x55555540108b <_Z11DisplayInfov> jmp 0x555555400f53 <main+265>	Checks if “userChoice” is equal to 1 If so calls DisplayInfo() function
mov 0x2013ab(%rip),%eax # 0x5555556022f4 <choice> cmp $0x2,%eax jne 0x555555400f53 <main+265> callq 0x555555401277 <_Z20ChangeCustomerChoicev> mov 0x20139b(%rip),%eax # 0x5555556022f4 <choice>	Checks if “userChoice” is equal to 2 If so calls ChangeCustomerChoice() function
cmp $0x3,%eax je 0x555555400f63 <main+281> jmpq 0x555555400e97 <main+77>	Checks if “userChoice is equal to 3 If so the application terminates If not the loop continues
mov $0x0,%eax pop %rbp retq	Returns 0 Terminates the main function

ChangeCustomerChoice function

Assembly Code Block	Explanation of Functionality

push %rbp mov %rsp,%rbp lea 0x346(%rip),%rsi # 0x5555554015c8 lea 0x200e17(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>	Outputs the prompt “Enter the number of the client you wish to change”
lea 0x201063(%rip),%rsi # 0x5555556022f8 <changechoice> lea 0x200f24(%rip),%rdi # 0x5555556021c0 <_ZSt3cin@@GLIBCXX_3.4> callq 0x555555400c60 <_ZNSirsERi@plt>	Takes integer input and stores value in variable “clientChoice”
lea 0x358(%rip),%rsi # 0x555555401600 lea 0x200df1(%rip),%rdi # 0x5555556020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0x555555400c90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>	Outputs the prompt “Please enter the client’s new service choice (1 = Brokerage, 2 = Retirement)”
lea 0x201041(%rip),%rsi # 0x5555556022fc <newservice> lea 0x200efe(%rip),%rdi # 0x5555556021c0 <_ZSt3cin@@GLIBCXX_3.4> callq 0x555555400c60 <_ZNSirsERi@plt>	Takes integer input and stores value in variable “serviceChoice”

mov 0x20102b(%rip),%eax # 0x5555556022f8 <changechoice> cmp $0x1,%eax jne 0x5555554012e0 <_Z20ChangeCustomerChoicev+105> mov 0x201024(%rip),%eax # 0x5555556022fc <newservice> mov %eax,0x200d82(%rip) # 0x555555602060 <num1> jmp 0x555555401342 <_Z20ChangeCustomerChoicev+203> mov 0x201012(%rip),%eax # 0x5555556022f8 <changechoice> cmp $0x2,%eax jne 0x5555554012f9 <_Z20ChangeCustomerChoicev+130> mov 0x20100b(%rip),%eax # 0x5555556022fc <newservice> mov %eax,0x200d6d(%rip) # 0x555555602064 <num2> jmp 0x555555401342 <_Z20ChangeCustomerChoicev+203> mov 0x200ff9(%rip),%eax # 0x5555556022f8 <changechoice> cmp $0x3,%eax jne 0x555555401312 <_Z20ChangeCustomerChoicev+155> mov 0x200ff2(%rip),%eax # 0x5555556022fc <newservice> mov %eax,0x200d58(%rip) # 0x555555602068 <num3> jmp 0x555555401342 <_Z20ChangeCustomerChoicev+203> mov 0x200fe0(%rip),%eax # 0x5555556022f8 <changechoice> cmp $0x4,%eax jne 0x55555540132b <_Z20ChangeCustomerChoicev+180> mov 0x200fd9(%rip),%eax # 0x5555556022fc <newservice> mov %eax,0x200d43(%rip) # 0x55555560206c <num4> jmp 0x555555401342 <_Z20ChangeCustomerChoicev+203> mov 0x200fc7(%rip),%eax # 0x5555556022f8 <changechoice> cmp $0x5,%eax jne 0x555555401342 <_Z20ChangeCustomerChoicev+203> mov 0x200fc0(%rip),%eax # 0x5555556022fc <newservice> mov %eax,0x200d2e(%rip) # 0x555555602070 <num5>	Checks if “clientChoice” is between 1 and 5 If so the service associated witht he provided client choice is replaced with “serviceChoice”

pop %rbp retq	Function epilogueg

CheckUserPermissonAccess Function

Assembly Code Block	Explanation of Functionality
push %rbp mov %rsp,%rbp push %rbx sub $0x48,%rsp mov %fs:0x28,%rax mov %rax,-0x18(%rbp)	Set up for the function
xor %eax,%eax lea -0x45(%rbp),%rax mov %rax,%rdi callq 0xd20 <_ZNSaIcEC1Ev@plt> lea -0x45(%rbp),%rdx lea -0x40(%rbp),%rax	Intitialize variables “username” and “password”
lea 0x588(%rip),%rsi # 0x1525 mov %rax,%rdi callq 0xce0 <_ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_@plt> lea -0x45(%rbp),%rax	Input/Output operations prompting the user to input their username and password and saving the input to the corresponding variable
mov %rax,%rdi callq 0xcb0 <_ZNSaIcED1Ev@plt> movl $0x0,-0x44(%rbp) lea 0x567(%rip),%rsi # 0x1526 lea 0x2010da(%rip),%rdi # 0x2020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>	Compares the value of “password” to the string “123” if equal returns true if not outputs “Invalid password. Please try again.” and restarts the loop

DisplayInfo Function

Assembly Code Block	Explanation of Functionality
push %rbp mov %rsp,%rbp	Set up for the function

lea 0x4c2(%rip),%rsi # 0x1558 lea 0x201003(%rip),%rdi # 0x2020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> mov %rax,%rdx mov 0x200f24(%rip),%rax # 0x201fd0 mov %rax,%rsi mov %rdx,%rdi callq 0xca0 <_ZNSolsEPFRSoS_E@plt> lea 0x4de(%rip),%rsi # 0x159c lea 0x200fdb(%rip),%rdi # 0x2020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> lea 0x200f3f(%rip),%rsi # 0x202010 <name1> mov %rax,%rdi callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> lea 0x4c0(%rip),%rsi # 0x15a0 mov %rax,%rdi callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> mov %rax,%rdx mov 0x200f6f(%rip),%eax # 0x202060 <num1> mov %eax,%esi mov %rdx,%rdi callq 0xd00 <_ZNSolsEi@plt> mov %rax,%rdx mov 0x200ecb(%rip),%rax # 0x201fd0 mov %rax,%rsi mov %rdx,%rdi callq 0xca0 <_ZNSolsEPFRSoS_E@plt> lea 0x49b(%rip),%rsi # 0x15b2 lea 0x200f82(%rip),%rdi # 0x2020a0 <_ZSt4cout@@GLIBCXX_3.4> callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> lea 0x200ef6(%rip),%rsi # 0x202020 <name2> mov %rax,%rdi callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt> lea 0x467(%rip),%rsi # 0x15a0 mov %rax,%rdi callq 0xc90 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@plt>	Creates a loop iterating through each client printing the information in the following format “’i+1’. ‘name[i]’ selected option ‘service[i]’” with and newline

nop pop %rbp retq	Function Epliogue