# CS 410 Project One: Proficiency Test

| main function | |
|---|---|
| **Assembly Code Block** | **Explanation of Functionality** |
| push %rbp<br>mov %rsp,%rbp<br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x17 <main+23> | Prints out welcome statement |
| callq 0x1c <main+28><br>mov %eax,0x0(%rip)<br>mov 0x0(%rip),%eax<br>cmp $0x1,%eax<br>je 0x40 <main+64><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x40 <main+64><br>mov 0x0(%rip),%eax | Loops and promts user to authenticate using CheckUserPermissionAccess until the user enter correct info |
| cmp $0x1,%eax<br>je 0x4d <main+77><br>jmp 0x17 <main+23> | Starts main body loop. |
| lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x60 <main+96><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x73 <main+115><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x86 <main+134><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0x99 <main+153><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0xac <main+172><br>lea 0x0(%rip),%rsi<br>lea 0x0(%rip),%rdi<br>callq 0xbf <main+191><br>mov %rax,%rdx<br>mov 0x0(%rip),%eax<br>mov %eax,%esi<br>mov %rdx,%rdi<br>callq 0xd2 <main+210><br>mov %rax,%rdx<br>mov 0x0(%rip),%rax | Displays menu with actions user can take |

| | |
|---|---|
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0xe7 <main+231> | |
| mov 0x0(%rip),%eax | |
| cmp $0x1,%eax | Select menu option 1 and calls DisplayInfo |
| jne 0xf9 <main+249> | |
| callq 0xf7 <main+247> | |
| mov 0x0(%rip),%eax | |
| cmp $0x2,%eax | Select menu option 2 and calls ChangeCustomerChoice |
| jne 0x109 <main+265> | |
| callq 0x109 <main+265> | |
| mov 0x0(%rip),%eax | |
| cmp $0x3,%eax | Selects menu option 3 and quits program |
| je 0x119 <main+281> | |
| jmpq 0x4d <main+77> | |
| mov $0x0,%eax | |
| pop %rbp | |

| CheckUserPermissonAccess function | |
|---|---|
| **Assembly Code Block** | **Explanation of Functionality** |
| push %rbp | |
| mov %rsp,%rbp | |
| push %rbx | |
| sub $0x48,%rsp | |
| mov %fs:0x28,%rax | Displays prompts, gets user input and compares password |
| mov %rax,-0x18(%rbp) | |
| xor %eax,%eax | |
| lea -0x45(%rbp),%rax | |
| mov %rax,%rdi | |
| callq 0x144 <_Z25CheckUserPermissionAccessv+36> | |
| lea -0x45(%rbp),%rdx | |
| lea -0x40(%rbp),%rax | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x15b <_Z25CheckUserPermissionAccessv+59> | |
| lea -0x45(%rbp),%rax | |
| mov %rax,%rdi | |
| callq 0x167 <_Z25CheckUserPermissionAccessv+71> | |
| movl $0x0,-0x44(%rbp) | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x181 <_Z25CheckUserPermissionAccessv+97> | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x194 <_Z25CheckUserPermissionAccessv+116> | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x1a7 <_Z25CheckUserPermissionAccessv+135> | |
| lea -0x40(%rbp),%rax | |
| mov %rax,%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x1ba <_Z25CheckUserPermissionAccessv+154> | |
| lea -0x40(%rbp),%rax | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x1cd <_Z25CheckUserPermissionAccessv+173> | |
| mov %eax,-0x44(%rbp) | |
| cmpl $0x0,-0x44(%rbp) | |
| jne 0x1dd <_Z25CheckUserPermissionAccessv+189> | Password correct, return 1 |
| mov $0x1,%ebx | |
| jmp 0x1e2 <_Z25CheckUserPermissionAccessv+194> | Password incorrect, return 2 |
| mov $0x2,%ebx | |
| lea -0x40(%rbp),%rax | |
| mov %rax,%rdi | |

| | |
|---|---|
| callq 0x1ee <_Z25CheckUserPermissionAccessv+206> | |
| mov %ebx,%eax | |
| mov -0x18(%rbp),%rcx | |
| xor %fs:0x28,%rcx | |
| je 0x23a <_Z25CheckUserPermissionAccessv+282> | |
| jmp 0x235 <_Z25CheckUserPermissionAccessv+277> | |
| mov %rax,%rbx | |
| lea -0x45(%rbp),%rax | |
| mov %rax,%rdi | |
| callq 0x210 <_Z25CheckUserPermissionAccessv+240> | |
| mov %rbx,%rax | |
| mov %rax,%rdi | |
| callq 0x21b <_Z25CheckUserPermissionAccessv+251> | |
| mov %rax,%rbx | |
| lea -0x40(%rbp),%rax | |
| mov %rax,%rdi | |
| callq 0x22a <_Z25CheckUserPermissionAccessv+266> | |
| mov %rbx,%rax | |
| mov %rax,%rdi | |
| callq 0x235 <_Z25CheckUserPermissionAccessv+277> | |
| callq 0x23a <_Z25CheckUserPermissionAccessv+282> | |
| add $0x48,%rsp | |
| pop %rbx | |
| pop %rbp | |
| retq | |

| DisplayInfo function | |
|---|---|
| **Assembly Code Block** | **Explanation of Functionality** |
| push %rbp | Display output header |
| mov %rsp,%rbp | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x258 <_Z11DisplayInfov+23> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x26d <_Z11DisplayInfov+44> | |
| lea 0x0(%rip),%rsi | Display output for first client |
| lea 0x0(%rip),%rdi | |
| callq 0x280 <_Z11DisplayInfov+63> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x28f <_Z11DisplayInfov+78> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x29e <_Z11DisplayInfov+93> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%eax | |
| mov %rdx,%rdi | |
| callq 0x2b1 <_Z11DisplayInfov+112> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x2c6 <_Z11DisplayInfov+133> | |
| lea 0x0(%rip),%rsi | Display output for second client |
| lea 0x0(%rip),%rdi | |
| callq 0x2d9 <_Z11DisplayInfov+152> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x2e8 <_Z11DisplayInfov+167> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x2f7 <_Z11DisplayInfov+182> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x30a <_Z11DisplayInfov+201> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |

| | |
|---|---|
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x31f <_Z11DisplayInfov+222> | |
| lea 0x0(%rip),%rsi | Display output for third client |
| lea 0x0(%rip),%rdi | |
| callq 0x332 <_Z11DisplayInfov+241> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x341 <_Z11DisplayInfov+256> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x350 <_Z11DisplayInfov+271> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x363 <_Z11DisplayInfov+290> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x378 <_Z11DisplayInfov+311> | |
| lea 0x0(%rip),%rsi | Display output for fourth client |
| lea 0x0(%rip),%rdi | |
| callq 0x38b <_Z11DisplayInfov+330> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x39a <_Z11DisplayInfov+345> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x3a9 <_Z11DisplayInfov+360> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x3bc <_Z11DisplayInfov+379> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x3d1 <_Z11DisplayInfov+400> | |
| lea 0x0(%rip),%rsi | Display output for fifth client |
| lea 0x0(%rip),%rdi | |
| callq 0x3e4 <_Z11DisplayInfov+419> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |

| | |
|---|---|
| callq 0x3f3 <_Z11DisplayInfov+434> | |
| lea 0x0(%rip),%rsi | |
| mov %rax,%rdi | |
| callq 0x402 <_Z11DisplayInfov+449> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%eax | |
| mov %eax,%esi | |
| mov %rdx,%rdi | |
| callq 0x415 <_Z11DisplayInfov+468> | |
| mov %rax,%rdx | |
| mov 0x0(%rip),%rax | |
| mov %rax,%rsi | |
| mov %rdx,%rdi | |
| callq 0x42a <_Z11DisplayInfov+489> | |
| nop | |
| pop %rbp | |
| retq | |

| ChangeCustomerChoice function | |
|---|---|
| **Assembly Code Block** | **Explanation of Functionality** |
| push %rbp | Prompt which client to change |
| mov %rsp,%rbp | |
| lea 0x0(%rip),%rsi | |
| lea 0x0(%rip),%rdi | |
| callq 0x444 <_Z20ChangeCustomerChoicev+23> | |
| lea 0x0(%rip),%rsi | Get user input for desired client |
| lea 0x0(%rip),%rdi | |
| callq 0x457 <_Z20ChangeCustomerChoicev+42> | |
| lea 0x0(%rip),%rsi | Prompt for new service choice for client |
| lea 0x0(%rip),%rdi | |
| callq 0x46a <_Z20ChangeCustomerChoicev+61> | |
| lea 0x0(%rip),%rsi | Get user input for desired choice |
| lea 0x0(%rip),%rdi | |
| callq 0x47d <_Z20ChangeCustomerChoicev+80> | |
| mov 0x0(%rip),%eax | Check if client 1, if so, assign new service choice |
| cmp $0x1,%eax | |
| jne 0x496 <_Z20ChangeCustomerChoicev+105> | |
| mov 0x0(%rip),%eax | |
| mov %eax,0x0(%rip) | |
| jmp 0x4f8 <_Z20ChangeCustomerChoicev+203> | |
| mov 0x0(%rip),%eax | |
| cmp $0x2,%eax | Check if client 2, if so, assign new service choice |
| jne 0x4af <_Z20ChangeCustomerChoicev+130> | |
| mov 0x0(%rip),%eax | |
| mov %eax,0x0(%rip) | |
| jmp 0x4f8 <_Z20ChangeCustomerChoicev+203> | |
| mov 0x0(%rip),%eax | |
| cmp $0x3,%eax | Check if client 3, if so, assign new service choice |
| jne 0x4c8 <_Z20ChangeCustomerChoicev+155> | |
| mov 0x0(%rip),%eax | |
| mov %eax,0x0(%rip) | |
| jmp 0x4f8 <_Z20ChangeCustomerChoicev+203> | |
| mov 0x0(%rip),%eax | |
| cmp $0x4,%eax | Check if client 4, if so, assign new service choice |
| jne 0x4e1 <_Z20ChangeCustomerChoicev+180> | |
| mov 0x0(%rip),%eax | |
| mov %eax,0x0(%rip) | |
| jmp 0x4f8 <_Z20ChangeCustomerChoicev+203> | |
| mov 0x0(%rip),%eax | |
| cmp $0x5,%eax | Check if client 5, if so, assign new service choice |
| jne 0x4f8 <_Z20ChangeCustomerChoicev+203> | |
| mov 0x0(%rip),%eax | |
| mov %eax,0x0(%rip) | |
| nop | |

| | |
|---|---|
| pop %rbp | |
| retq | |