CS540: Foundations of Computing System Design Lab II Report

Submitted by-

Name: Prem Swarup Roll No: 2003318

Branch: Mathematics & Computing (2024 batch)

Given code file attached separately for calendar.cpp, in which I try to make a calendar for Sept of september and add an event at some date in the calendar, while adding the event following line(line no. 116) is getting called:

Sept.addEvent(date, event1);

where, "Sept" is object for september month,

"addEvent" is function or method of the "Sept" object

"date" is int variable containing date

"event1" is an event object

and I can see in disassembly view following instruction on registers are being processed:

7/1000033333337070		
9x900055555557082	48 8d 95 e0 fe ff ff	lea rdx,[rbp-0x120]
0×000055555557089	48 8d 85 20 ff ff ff	lea rax,[rbp-0xe0]
0x000055555557090	48 89 d6	mov rsi,rdx
9×900055555557093	48 89 c7	mov rdi,rax
0x0000555555557096	e8 39 0a 00 00	call 0x55555557ad4 < ZN5EventC2ERKS_>
9x90005555555709b	8b 8d /c fe ff ff	mov ecx,DWORD PIR [rbp-0x184]
0x00005555555570a1	48 8d 95 20 ff ff ff	lea rdx,[rbp-0xe0]
0x0000555555570a8	48 8d 85 60 ff ff ff	lea rax,[rbp-0xa0]
0x00005555555570af	89 ce	mov esi,ecx
9x9000555555570b1	48 89 c7	mov rdi,rax
9x9000555555570b4	e8 cd f9 ff ff	call 0x55555556a86 <_ZN13monthCalendar8acdEventEi5Event>
0x0000555555570b9	48 8d 85 20 ff ff ff	lea rax,[rhp-0xe0]
0×00005555555570c0	48 89 c7	mov rdi,rax
D 0x00005555555570c3	e8 dc 09 00 00	call 0x55555557aa4 <_ZN5EventD2Ev>

Fig.1: Disassembly lines corresponding to code on line no. 116 as given before

Line by line brief about above lines of instructions on registers are as follows:

- 1. lea rdx, [rbp-0x120]:
 - Calculates an address to be used, possibly for storing the 'Event' object, and stores it in 'rdx'.
- 2. lea rax, [rbp-0xe0]:
- Likely computes an address related to the 'unordered_map' within the 'Sept' object, and stores it in 'rax'.
- 3. mov rsi, rdx:
 - Copies the address calculated in step 1 to 'rsi', possibly preparing it for a function call.
- 4. mov rdi. rax:
 - Copies the address calculated in step 2 to 'rdi', likely preparing it for a function call.

- 5. call 0x555555557ad4 <_ZN5EventC2ERKS_>:
- Calls a constructor, possibly for an 'Event' object. It may be related to creating an event to be inserted into the 'unordered_map'.
- 6. mov ecx, DWORD PTR [rbp-0x184]:
 - Retrieves a 32-bit value from memory, possibly for further operations.
- 7. lea rdx, [rbp-0xe0]:
 - Recalculates an address related to the 'unordered_map'.
- 8. lea rax, [rbp=0xa0]:
 - Calculates an address, possibly related to the 'Sept' object.
- 9. mov esi, ecx:
 - Copies the value retrieved earlier into 'esi', likely preparing it for a function call.
- 10. mov rdi, rax:
 - Copies the address calculated in step 8 to 'rdi', possibly preparing it for a function call.
- 11. call 0x555555556a86 <_ZN13SeptCalendar8addEventEi5Event>:
 - Calls a function, possibly for adding an event to the 'unordered_map' within the 'Sept' object.
- 12. lea rax, [rbp-0xe0]:
 - Recalculates an address related to the 'unordered_map'.
- 13. mov rdi,rax:
 - Copies the address calculated in step 12 to 'rdi', likely preparing it for a function call.
- 14. call 0x55555555aa4 <_ZN5EventD2Ev>:
 - Calls a destructor, possibly for the 'Event' object that was created earlier.

And final memory address that I obtained using '&' operator on variable holding event in 'Sept' object is: 0x55555556ec0

where, 0x5555556ec0 is a memory address in hexadecimal notation. It represents a location in the computer's memory where the Event object is finally stored.