

CMSC216: Practice Exam 2A

Fall 2025

University of Maryland

Exam period: 20 minutes Points available: 40 Weight: 0% of final grade

Problem 1 (20 pts): Nearby is a `main()` demonstrating the use of the function `setday()`. Below each call to `setday()`, its expected behavior and return are printed.

Implement this function according to the documentation given in **x86-64 assembly**. Comments below the `yearday_t` struct give information about how it lays out in memory and as a packed register argument.

```
.text
.global setday
setday:  ## YOUR CODE BELOW ##
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // global: days since 1/1/1970
5 int DAYS_SINCE_1970 = 0;
6
7 // struct containing date info
8 typedef struct {
9     int day;    // current day in this year
10    int year;    // current year
11 } yearday_t;
12 // Layout of yearday_t in memory and
13 // as a packed register argument.
14 //
15 // |           | Byte |   Byte | Packed |
16 // | Field | Size | Offset | Bits |
17 // |-----+-----+-----+-----|
18 // | day   | 4   |    +0  | 0-31  |
19 // | year  | 4   |    +4  | 32-63 |
20
21 int setday(yearday_t yd);
22 // DEFINED IN ASSEMBLY
23 //
24 // Extracts the day and year fields
25 // from the provided struct yd. If day is
26 // negative or year is < 1970, sets the
27 // global variable DAYS_SINCE_1970 to
28 // be -1 and returns 1. Otherwise
29 // computes the number of days since
30 // 1970 based on these fields assuming
31 // NO LEAP YEARS and 365 days per
32 // year. Sets the global variable
33 // DAYS_SINCE_1970 to this value and
34 // returns 0.
35
36 int main(int argc, char *argv[]){
37     // Demonstrate 3 examples of setday()
38     int ret;
39     yearday_t yd1 =
40         { .day = 20, .year = 1970 };
41     ret = setday(yd1);
42     printf("%3d days since 1970 (ret: %d)\n",
43           DAYS_SINCE_1970, ret);
44     // 20 days since 1970 (ret: 0)
45
46     yearday_t yd2 =
47         { .day = 3, .year = 1972 };
48     ret = setday(yd2);
49     printf("%3d days since 1970 (ret: %d)\n",
50           DAYS_SINCE_1970, ret);
51     // 733 days since 1970 (ret: 0)
52
53     yearday_t yd3 =
54         { .day = 7, .year = 1955 };
55     ret = setday(yd3);
56     printf("%3d days since 1970 (ret: %d)\n",
57           DAYS_SINCE_1970, ret);
58     // -1 days since 1970 (ret: 1)
59     return 0;
60 }
```

Problem 2 (10 pts): Below is a `main()` function which uses the function `setarray()`. As the demo shows, compiling with a C version of this function works fine but the assembly version has some problems.

```
// setarray_main.c          // setarray_c.c          ## setarray_asm.s
#include <stdio.h>
int main(){
    long arr[3];
    setarray(arr,3,10);
    for(int i=0; i<3; i++){
        printf("%2d ",arr[i]);
    }
    printf("\n");
    return 0;
}

1 void setarray(long *arr,
2               long len,
3               long val)
4 {
5     for(long i=0; i<len; i++){
6         arr[i] = val;
7     }
8     return;
9 }

1 .text
2 .globl setarray
3 setarray:
4     movq    $0, %rax
5 .LOOP:
6     cmpq    %rsi, %rax
7     jg      .DONE
8     movq    %rdx, (%rdi,%rax,8)
9     addq    $1, %rax
10    jmp     .LOOP
11 .DONE:
12    ret

>> gcc setarray_main.c setarray_c.c          >> gcc setarray_main.c setarray_asm.s
>> ./a.out                                    >> ./a.out
10 10 10                                     10 10 10
*** stack smashing detected ***: terminated
Aborted (core dumped)
```

Describe why the assembly version causes Stack Smashing and how to fix it.

Problem 3 (10 pts): While debugging a binary program, Nils Punters encounters an assembly instruction that baffles him: `test %rax,%rax`. Nils is struggling to understand what this could possibly accomplish. **Explain** what the `testX` instruction does AND what it is likely being used to do in the code Nils is examining which is shown nearby.

```
+=====GDB=====+
|>>0x55154 <nodes_sorted+11> test    %rax,%rax # Nils: WTF? |
| 0x55157 <nodes_sorted+14> je      0x55555555167 <nodes_sorted+30> |
| 0x55159 <nodes_sorted+16> mov     (%rax),%edx |
| 0x5515b <nodes_sorted+18> cmp     %ecx,%edx |
| 0x5515d <nodes_sorted+20> jl      0x55555555173 <nodes_sorted+42> |
| 0x5515f <nodes_sorted+22> mov     0x8(%rax),%rax |
| 0x55163 <nodes_sorted+26> mov     %edx,%ecx |
| 0x55165 <nodes_sorted+28> jmp     0x55555555154 <nodes_sorted+11> |
| 0x55167 <nodes_sorted+30> mov     $0x1,%eax |
| 0x5516c <nodes_sorted+35> ret     |
+=====+

```