13 Lecture: C Quiz Post-Mortem

Outline:

Announcements
    The Exam
Discussion of the Exam
Selected Solutions
Aside: Software Reliability and Getting it right

13.1 Announcements

- Coming attractions:

Event	Subject	Due Date			Notes
asgn3	hencode/hdecode	Fri	Nov 3	23:59	
lab05	mypwd	Mon	Nov 6	23:59	
asgn4	mytar	Mon	Nov 27	23:59	
asgn5	mytalk	Fri	Dec 1	23:59	
lab07	forkit	Mon	Dec 4	23:59	
asgn6	shell	Fri	Dec 8	23:59	

  Use your own discretion with respect to timing/due dates.

- Notes:
  - It’s ’\0’ not ’/0’
  - NULL vs. nul (’\0’) not ’/0’
  - `A[i] vs.  *(A + i)`. The latter is weird.
- MAX CODE is 255 bits
- asgn2: 67 submissions, 51 built

13.1.1 The Exam

- fear not expressions. I.e. Don’t do:

```
Atemp = s[i];
Btemp = t[i];
if ( Atemp == Btemp )
```

- No: *(s+i)
- Read the question!
- How to turn O(n) into O(n^2):

```
 for (s=string; *s; s++)
vs.
for(i=0; i < strlen(string); i++ )
vs.
```

```
len = strlen(string)
     for(i=0; i < len; i++ )
```

This is an argument for “`const char *s`”

- add strlen() to the soln set.
- Don’t make things harder than they need to be.
- sizeof() vs. strlen()
- lseek(2)

C Quiz		
High	60.0	100.0%
Low	21.0	30.0%
Mean	39.4	70.9%
Median	38.0	71.7%
S.D.	9.6	14.3%

Grade	**Cutoff**	**Percent**
Min A−	52.5	(87.5%)
Min B−	45.0	(75.0%)
Min C−	37.5	(62.5%)
Min D−	30.0	(50.0%)

13.2 Discussion of the Exam

Final thought: This was not a particularly difficult exam. In most cases what I asked you to do was either a fairly simple example of a concept or something you’d done before.

13.3 Selected Solutions

We discussed some things. (Read!)

Notes:

- C99 vs. ANSI C:
  - decls and code may not be mixed
  - variable-sized arrays do not exist
- Keep it as simple as possible.
- The type is “struct node_st”, etc.
- **Read the exam**

Problem Notes:

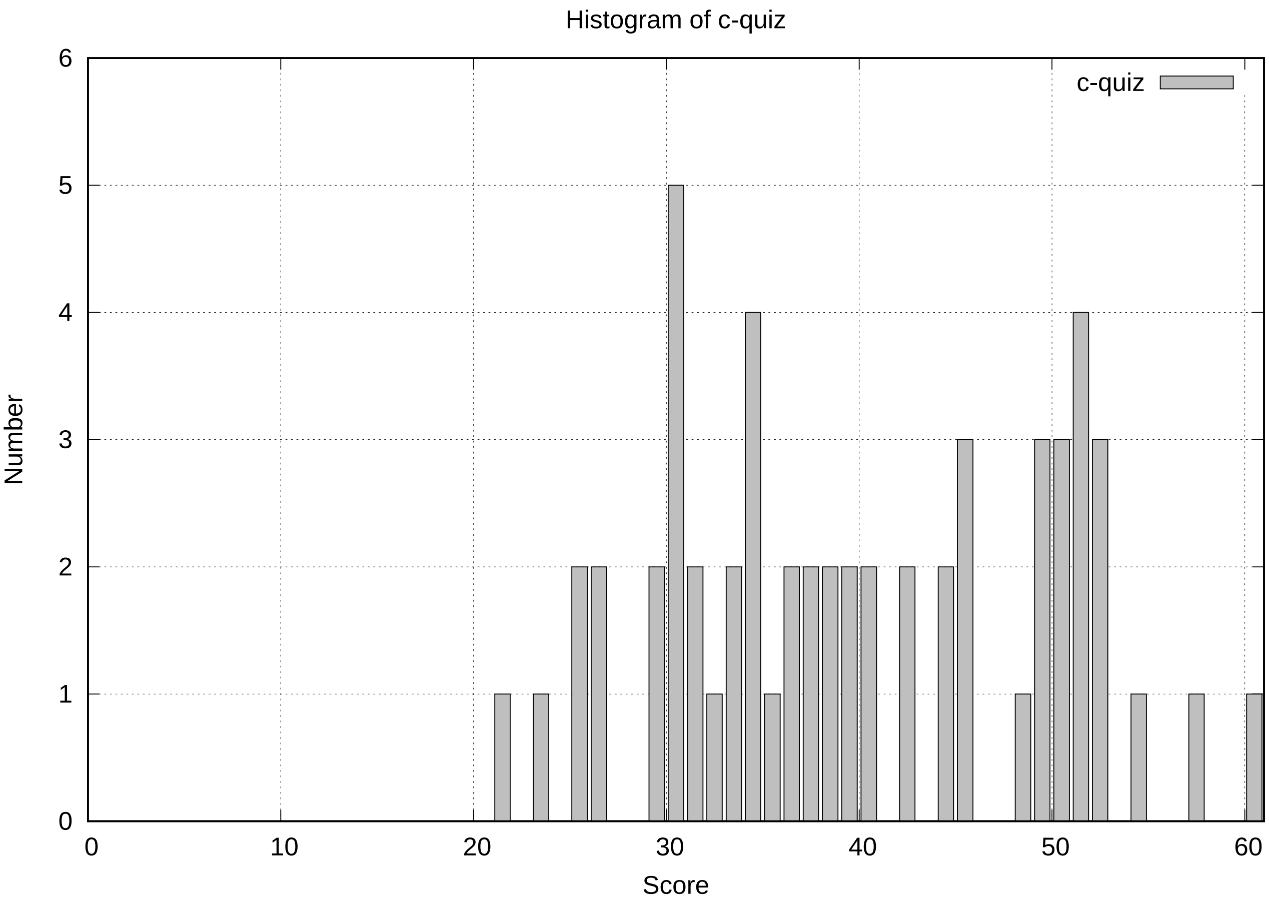

Figure 63: Histogram of scores for the quiz

1. The macro version (putc) might allow for more efficient execution because it will be expanded in place avoiding the overhead of a function call, but a macro expansion might evaluate its argument more than once making it an inappropriate choice in a situation where one of its arguments might have side-effects. Consider the effect such an expansion with multiple evaluation would have on:

   ```
   while ( *s ) { putc(s[i++],stdout) ...
   ```

2. If the given header is included by more than one source file, there would be a linker error because of multiply defined symbols.

3. mean(): overflow and types

4. strchr()

   - Remember the special behavior with nul
   - '\0' is not NULL
   - Check for NULL

5. merge_sorted

   -

13.4 Aside: Software Reliability and Getting it right

It's harder than it looks.

A class exercise in getting it right. Consider `rand()`

How do you take a value $r \in [0, \text{RAND\_MAX}]$ and map it onto $[A, B]$.

remainder	uneven
$r' = A + \left\lfloor \frac{r}{\text{RAND\_MAX}} \times (B - A) \right\rfloor$	Also uneven (only one value could give you B)
$r' = A + \left\lfloor \frac{r}{\text{RAND\_MAX}} \times (B - A + 1) \right\rfloor$	Now you can get B+1!
$r' = A + \left\lfloor \frac{r}{\text{RAND\_MAX}+1} \times (B - A + 1) \right\rfloor$	Works, but what if RAND_MAX is $2^{31} - 1$?

Double has 53 bits of precision.