Complex C Types
Welcome!

Today:  
➔ Complex C types  
➔ Debugging

Logistics:  
➔ Assign 2 up by next class.  
➔ 99% started on Assign 1, 77% submitted.  
➔ Prefer Piazza, then email: s23-comp-211-002-staff-cs @cs.unc.edu

Did you know…

```c
char fun_message[] = {0x54, 0x72, 0x79, 0x20, 0x72, 0x75, 0x6e, 0x6e, 0x69, 0x6e, 0x67, 0x20, 0x2e, 0x2f, 0x74, 0x65, 0x74, 0x72, 0x69, 0x73, 0x20, 0x6f, 0x6e, 0x20, 0x74, 0x68, 0x65, 0x72, 0x00};
```

It seems like very few of you got through this last time, so here it is again.

See the slides from lecture 4 for the ASCII table.
Complex C Types

Let's use them to control the NVIDIA GPU scheduler...
Complex C Types: Enum

nvdebug.h, enum
ENTRY_TYPE

Code available at
Complex C Types: Bit-packed Struct

nvdebug.h, struct
gv100_runlist_tsg

Code available at
Complex C Types: Union

type runlist_base_t

nvdebug.h, runlist_base_t

Code available at
Complex C Types: Everything All at Once!

nvdebu(p+dbug.h, runlist_info_t

Code available at
Complex C Types

Your turn!

Try it yourself!

```bash
$ wget https://www.cs.unc.edu/~jbakita/teach/comp211-s23/l6/mystery2.c
$ gcc mystery2.c -o myst
$ ./myst
```

```c
#include <stdio.h>

struct char_info {
    char __misc:5;
    char is_printable:2;
    char is_extended:1;
};

union e_char {
    char c;
    struct char_info i;
};

int main() {
    union e_char ch;
    int my_str_len = 16;
    char my_str[] = \"\0mHello world\";

    for (int i = 0; i < my_str_len; i++) {
        ch = (union e_char)my_str[i];

        printf("char '\%x' at index \%d
", ch.c, i);
        if (ch.i.is_extended) {
            printf("does not have a well-agreed upon meaning, and ");
        } else {
            printf("has a well-agreed upon meaning, and ");
        }
        if (ch.i.is_printable) {
            printf("is printable.\n');
        } else {
            printf("is not printable.\n');
        }
    }
    return 0;
}
```

What will this print for `i = 0`?

https://PollEv.com/joshuabakita182

Grab these slides from the website to see the text up close.
Recommended Readings

Structs:
- See readings from last time.

Unions:
- The GNU C Reference Manual, §2.3 (3 pgs)
  - info gnu-c "Unions"
- The C Programming Language, §6.8 (2 pgs)

Enumerations:
- The GNU C Reference Manual, §2.2 (1 pg)
  - info gnu-c "Enumerations"
- The C Programming Language, §2.3 (3 pgs)
Thanks!

Questions?

See office hour calendar on the website for availability.

Assignment 1 due tonight!

Contact:
Email: hacker@unc.edu
Twitter: @JJBakita
Web: https://cs.unc.edu/~jbakita