Week 04M: When In C..

Thuy Vu

Reminders

- Assignment 4
  - `web.cs.ucla.edu/classes/winter17/cs35L/assign/assign4.html`
  - Time due 23:55 this Saturday, February 04

to ‘C’ or not to ‘C’

1. C can produce truly tiny executable. C++ might introduce undesirable overhead.
2. C does not support function overloading. C++ does.
3. http://thread.gmane.org/gmane.comp.version-control.git/57643/focus=57918
4. http://programmers.stackexchange.com/questions/113295/when-to-use-c-over-c-and-c-over-c

1. basic data types: `int`, `float`, `double`, `char`, `void`... but no `bool`
2. pointers: `int *ptr; int var=77; ptr=&var;`
3. dereferencing: `double x, *ptr; ptr=&x; *ptr=7.8;`
4. pointer-pointer: `char c='A'; char *c_ptr=&c; char **c_ptr_ptr=&c_ptr;`
5. function-pointer → functor: an object to be called like a function
   - `double (*func_ptr) (double, double);`
   - `func_ptr=&pow; // points to function pow()`
   - `double result=(*func_ptr)(1.5, 2.0); // call referenced by functor`
   - `result=func_ptr(1.5, 2.0); // call using function call`
6. functor example
   - `int compare(const void *a, const void *b)`
     `{ return *(int*)a - *(int*)b;}`
   - `...`
   - `int values[]={40, 50, 10, 20, 60};`
   - `qsort(values, 6, sizeof(int), compare);`
7. structure
8. memory management
   - `void *malloc(size_t size); // allocates size bytes`
   - `void *realloc(void *ptr, size_t size); // changes size for ptr`
   - `void free(void *ptr); // frees allocated memory for ptr`
9. I/O: `fprintf`, `fscanf`, `getchar`, `putchar`
10. Compile: `gcc -o hello_binary -g hello_code.c`
    - gcc, not g++
    - -g to includes debugging info

knowing how to debug is a must

1. How: reproduce the bug → track down the bug's origin → fix it
2. Debugger: stepping, breaking, reading memory
3. GNU Debugger – `gdb`

GDB

1. Compile: `gcc [...] -g [...]`
2. Debug: `gdb <executable>` (or in gdb, use `file <excutable>`)
3. Run: in gdb, use `run [arguments]`
4. Breakpoint: in gdb, use:
   - `break file.c:6` // break at
   - `break my_function` // break when function is called
   - `break [position] if <expression>` // break if condition
   - `info breakpoints /break/br/b` // get the info
   - `delete`, `disable`, `enable`, `ignore`
   - at breakpoint, `c`-ontinue, `n`-next, `s`-tep, `f`-inish
5. Display: in gdb, use:
   - `print [/format] <expression>` // format: d-x-o-t
6. Watch: in gdb, use:
   - `watch my_var`
   - `rwatch <expression>`

Lab

1. Download and install the program
2. Download and apply the patch
   - Find and explain the error
3. Reproduce the bug
   - Unix time (aka POSIX time, Epoch time) is #seconds elapsed since Thursday, 1 January 1970
   - `df -T`
   - `ext4` provides timestamps measured in nanoseconds
   - `nfs` (network file system)
4. Debug with GDB
   - set break where suspicious
5. Create patch and test

Exercise

1. exclusive-ORing 42
2. comparison function
3. `int main(...)`
   - `getchar()`