CSE 333 – Section 7

Midterm Q&A, Symbol Table and Sort a Vector

Andrew Davies Johnny Yan

Department of Computer Science & Engineering
University of Washington

November 7, 2013

Midterm Q&A

You did a pretty good job. But any questions?

Sample Code

example.c

```c
int N = 20;

char toUpper(char c);
void printResult(char c, int count);

int max(int x, int y) {
  return x > y ? x : y;
}

int main(int argc, char *argv[]) {
  int count = 0;
  char c = 0;
  for ( char **argPtr = argv; *argPtr; argPtr++ ) {
    for ( const char* p = *argPtr; *p; p++ ) {
      c = max(c, toUpper(*p) );
      count++;
    }
  }
  printResult(c, count);
  return 0;
}

char toUpper(char c) {
  return c & ~0x20;
}
```

Object Dump

```
$ objdump -t example.o

example.o:     file format elf64-x86-64

SYMBOL TABLE:
0000000000000000 l    df *ABS* 0000000000000000 example.c
0000000000000000 l    d  .text 0000000000000000 .text
0000000000000000 l    d  .data 0000000000000000 .data
0000000000000000 l    d  .bss 0000000000000000 .bss
0000000000000000 l    d  .note.GNU-stack 0000000000000000 .note.GNU-stack
0000000000000000 l    d  .eh_frame 0000000000000000 .eh_frame
0000000000000000 l    d  .comment 0000000000000000 .comment
0000000000000000 g     O .data 0000000000000004 N
0000000000000000 g     F .text 0000000000000016 max
0000000000000016 g     F .text 0000000000000091 main
00000000000000a7 g     F .text 0000000000000012 toUpper
0000000000000000         *UND* 0000000000000000 printResult
```

C++ STL

C++ standard-library

- defines container classes
- defines generic algorithms
- many more http://www.cplusplus.com/reference
- take care of memory management

Containers

Sequential

- vector
- list
- deque
- stack
- queue/priority_queue
- array(C++11)
- forward_list(C++11)

Associative

- map/multimap
- set/multiset
- unordered_map/unordered_multimap(C++11)
- unordered_set/unordered_multiset(C++11)

Containers cont.

Containers

- Each container, except adaptors, defines *begin* and *end* functions yield iterators
- Using dereference operator(*) to access the elements
- Using increment operator(++) to advance to the next element
- Some operations on the container may invalidate the iterator

Algorithms

2nd level generic

- operate on different types of containers
- various element types

general forms

- *alg(beg, end[, other parms])*
- *alg(beg, end, dest[, other parms])*
- *alg(beg, end, beg2[, other parms])*
- *alg(beg, end, beg2, end2[, other parms])*

vector & sort

vector

```
#include <vector>
  // statements
  std::vector<int> vi;
  vi.push_back(3);
  // statements
```

algorithm

```
#include <algorithm>
  // statements
  std::sort(vi.begin(), vi.end());
  // statements
```