Some Basic & C Programming Review

Hengtong Zhang
E-mail: hengtong@buffalo.edu
CSE 489 Information

Instructor: Dr. Lu Su
Office: 321 Davis Hall
Email: lusu [at] buffalo.edu,

Office Hours
Mon. and Wen. 2:00 to 3:00 PM

TA: Wenjun Jiang, Hengtong Zhang

Course website:
https://www.cse.buffalo.edu/~lusu/cse4589/
Office Hours

Wenjun Jiang, Hengtong Zhang

Office: 302 Davis Hall/300 Davis Hall Student Lounge
Office Hours: Tue. 4:00 to 5:00 PM Thur. 1:30 to 2:30 PM
Email: wenjunji [at] buffalo.edu  hengtong [at] buffalo.edu
Recitation Hours

Rec 1 - Wednesday -> 3:00-3:50 PM (Baldy 112)
Rec 2 - Tuesday -> 8:00-8:50 AM (CANCELED)
Outline

Part 1: Some basics

Part 2: C Programming Language Review
Log into CSE Servers

• ssh <ubit_name>@timberlake.cse.buffalo.edu

• Some other useful commands:
  A) Copy locally – cp
  B) Copy across servers – scp

• Basic questions:
  Can we use putty on PC – Yes
  Can we login using Mac – Yes
  Can we write our code in local machines – Yes. But you **must** test on CSE servers. We grade on that.
SCP Commands

Copy the file "foobar.txt" from a remote host to the local host

$ scp your_username@remotehost.edu:foobartxt /some/local/directory

Copy the file "foobartxt" from the local host to a remote host

$ scp foobartxt your_username@remotehost.edu:/some/remote/directory

Copy the directory "foo" from the local host to a remote host's directory "bar"

$ scp -r foo your_username@remotehost.edu:/some/remote/directory/bar

Copy the file "foobartxt" from remote host "rh1.edu" to remote host "rh2.edu"

$ scp your_username@rh1.edu:/some/remote/directory/foobartxt your_username@rh2.edu:/some/remote/directory/
CSE Servers You Use

• Timberlake
• Highgate
• Euston
• Embankment
• Underground
How to Compile

For C programs

use gcc

e.g., gcc hello_world.c –o hello-world
C Strings

● Implemented with sequence of characters (char)
  
  char *message = "This is a string literal."

● Null termination ('\0') is needed for String literal

● Two primary ways to work with strings
  
  String Literals
  
  Char arrays

● String Literals ARE null terminated

● Char arrays are automatically NOT null terminated

● Char arrays equated to string literals ARE null terminated
String Literals

- Generally stored in read-only area
- Not a good idea to modify them
- Avoid passing them to functions

```c
char *string = "CSE489"; //null terminated
string[3] = '5'; //Undefined behavior
```
Char Arrays

- Sequence of char literals
- It is OK to modify them
- It is OK to pass them to functions

```c
char a[7]; // NOT null terminated
char a[7] = "CSE489"; // null terminated
char a[] = "CSE489"; // null terminated
char a[6] = "CSE489"; // NOT null terminated
```
Functions Dealing with Strings

Please refer to the website:

http://en.cppreference.com/w/c/string/byte
void swap2(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
    assert(*a == 17);
    assert(*b == 42);
    // they're swapped!
}

int x = 42;
int y = 17;
swap2(&x, &y); // give the function pointers to our variables
assert(x == 17);
assert(y == 42);
// yes, they're swapped!
Malloc() and Memcpy()

Malloc() – Allocate requested memory

Syntax - void *malloc(size_t size)

e.g., - char *str = (char *) malloc(15);

Memcpy() – Copies ‘n bytes’ from one memory to another.

Syntax - void *memcpy(void *s1, const void *s2, size_t n)

s1 -- This is pointer to the destination array

s2 -- This is pointer to the source

n -- This is the number of bytes to be copied.
Debug C code

Debug with GDB (GNU Debugger)
Debug with IDE (Extremely convenient, e.g. Clion, Visual Studio with Linux Extension)
Debug with printf (Best for multi-machine programs)
Tips

Project is a bit complicated. Do unit test once finishing each module.

Beej network tutorial can be very useful.

Use the most straightforward way to finish the task.
Acknowledgements

Some slides contain material developed and copyrighted by Jingyuan Fan (SUNY Buffalo).