The first recitation of CSCI 3753

zhiyuan.liu@colorado.edu

Today's topic

- (1) Building development environment
- (2) Requirement on your code
- (3) Requirement on the Linux
- (4) Some basic of C language and data structure
- (5) Policy for grading
- (6)Office Hour

Building development environment

- VirtualBox(free) + VM(Spring 2016 Edition)
- Strongly suggestion to use VM.(Parameters,Rights,Problems,Crash Down)
- Download the VirtualBox, install.(official website)
- https://foundation.cs.colorado.edu/vm/
- Can’t remember. Google CU CS VM
- Download the VM.

Import VM Image

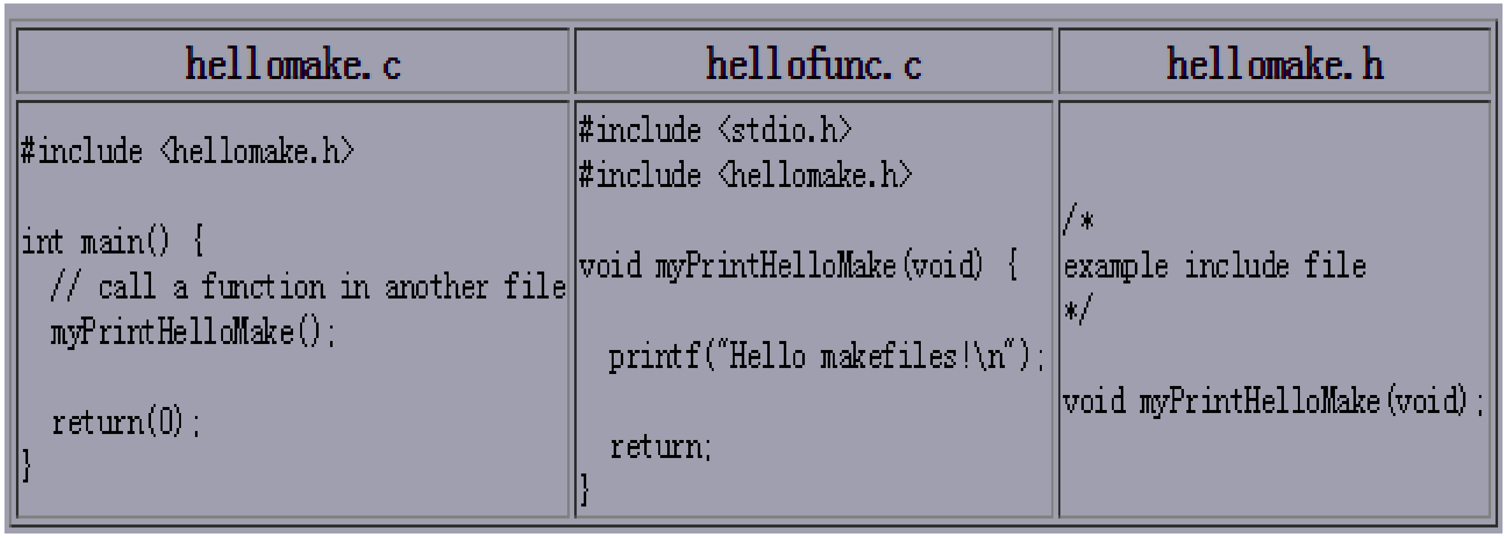
- Launch VirtualBox
- select File > Import Appliance...
- select the *.ova VM image
- Appliance Settings screen, check the Reinitialize the MAC address...
- Wait to finish importing
- All the instructions can be obtained from the website.

Requirement on your code

- Comments(very important, not long but essential)
- Well indentation(key style of code)
- Well named parameters.(easy to understand , chose one style you like)
- Well defined function(some students even didn’t create function, good to reuse)
- README(file’s name , how to run, the main idea and functions of your code, even your bugs).

Requirement on your code

- Makefile(Suggested but not required)
- Very convenient if you need to compile many files(many .c .h, other library you want to use)
- A simple example:



hellomake.c

```
#include <hellomake.h>

int main() {
  // call a function in another file
  myPrintHelloMake();

  return(0);
}
```

hellofunc.c

```
#include <stdio.h>
#include <hellomake.h>

void myPrintHelloMake(void) {

  printf("Hello makefiles!\n");

  return;
}
```

hellomake.h

```
/*
example include file
*/

void myPrintHelloMake(void);
```

Makefile

- gcc -o hellomake hellomake.c hellofunc.c -I.

- hellomake: hellomake.c hellofunc.c

  gcc -o hellomake hellomake.c hellofunc.c –I.

Makefile

- CC=gcc
- CFLAGS=-I.
- hellomake: hellomake.o hellofunc.o

```
	$(CC) -o hellomake hellomake.o hellofunc.o -I.
```

But if the .h file changes…

Makefile

- CC=gcc
- CFLAGS=-I.
- DEPS = hellomake.h
- %.o: %.c $(DEPS)

  $(CC) -c -o $@ $^ $(CFLAGS)

  hellomake: hellomake.o hellofunc.o

  gcc -o hellomake hellomake.o hellofunc.o -I.

  What if we have many .h files, locate in different paths?

Makefile

- CC=gcc
- CFLAGS=-I. -I /usr/local/include
- DEPS = hellomake.h ….
- OBJ = hellomake.o hellofunc.o
- %.o: %.c $(DEPS)

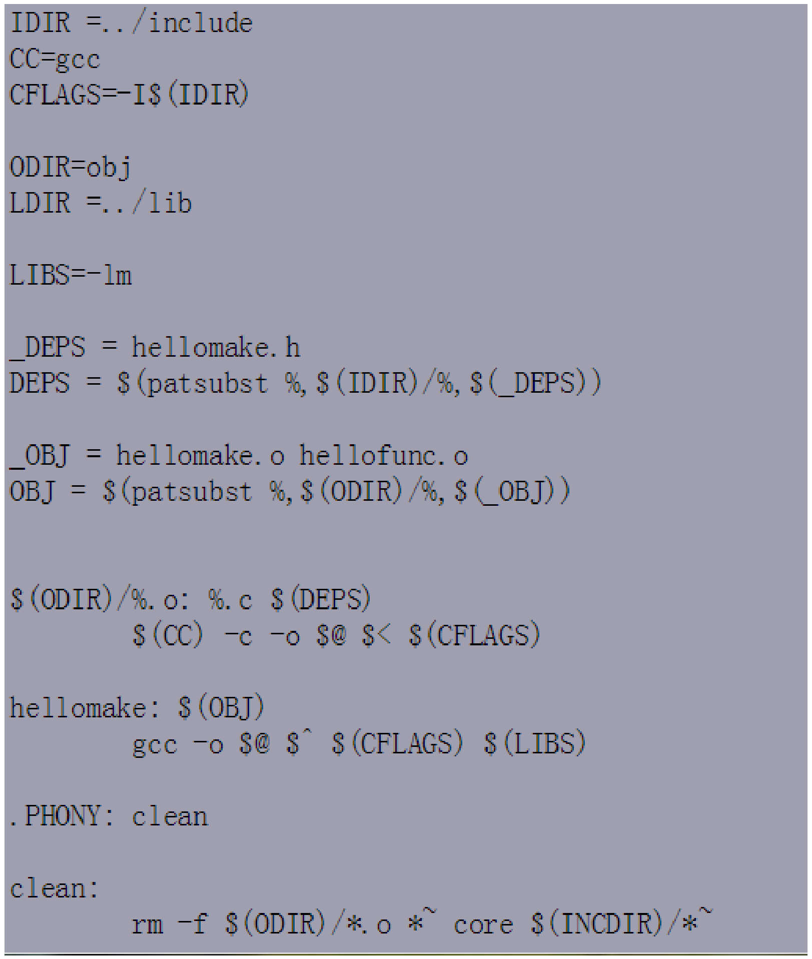
  $(CC) -c -o $@ $^ $(CFLAGS)

  hellomake: $(OBJ)

  gcc -o $@ $^ $(CFLAGS)

  But if we want to use link to use other lib.-lm – pthread? Make clean?

Makefile



```
IDIR =../include
CC=gcc
CFLAGS=-I$(IDIR)

ODIR=obj
LDIR =../lib

LIBS=-lm

_DEPS = hellomake.h
DEPS = $(patsubst %,$(IDIR)/%,$(_DEPS))

_OBJ = hellomake.o hellofunc.o
OBJ = $(patsubst %,$(ODIR)/%,$(_OBJ))


$(ODIR)/%.o: %.c $(DEPS)
	$(CC) -c -o $@ $< $(CFLAGS)

hellomake: $(OBJ)
	gcc -o $@ $^ $(CFLAGS) $(LIBS)

.PHONY: clean

clean:
	rm -f $(ODIR)/*.o *~ core $(INCDIR)/*~
```

This leaves for you.
patsubst. Make clean.
This example from:

http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/

More help:
GNU make manual.

Requirement on your code

- Take care of your code…..
- Back up your project asap.
- Computer crashed down can not be an excuse.
- Git,bitbucket…
- OR: use dropbox.
- In your VM, dropbox have been installed.

- ln -s /path/to/desired-folder ~/Dropbox/desired-folder

Requirement on the Linux

- Basic linux command:

- cat,cd,cp,mkdir,rm,ln,ls,more,less,mv,pwd,vim,
- Chmod(r,w,x),uname,sudo,history,apt,update,
- tar,man,alias,jobs,top,kill,make,ping,ps,sleep,
- time,which,whereis....

Requirement on the Linux

- Also, we need you to be familiar with one code editor.

- Vim,Emacs,Codeblocks,Eclipse,gedit any IDEs for c or c++.

- Lean to write simple scripts and run scripts.
- #!/bin/bash
- echo "hello, $USER.
- I wish to list some files of yours"
- echo "listing files in the current directory, $PWD"
- ls # list files

Requirement on the Linux

- Learn how to install opensource softwares from GNU(one program we will use openssh etc.)
- ./configure --prefix=
- make
- make test sometimes need make lib
- sudo make install
- Generally, the default headfiles are in /usr/local/include,
- The default librarys are in /usr/local/lib

Requirement on the Linux

- Learn how to solve errors you face.
- If you begin to use linux or new user, errors are always everywhere.
- Ask Ubuntu,stackflows.
- Understand the error information,try you best.
- No permitions , no such files…
- Copy the error information, google it, usually you can find the solution.

Some basic of C language and data structure(basic requirements)

- Input/output
- File IO(read,write,rewind)
- Array
- C pointer
- Structure
- Linked list(create,add,remove,search)
- Memory allocation(malloc,realloc,free)

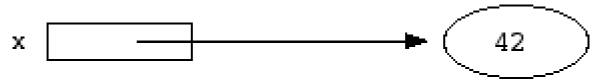
Pass by value && Pass by reference

- Compare:
- void swap(int num1, int num2) {

```
    int temp = num1;
    num1 = num2;
     num2 = temp;
     }
    void swap(int& num1, int& num2) {
    int temp = num1;
    num1 = num2;
    num2 = temp;
    }
```

C pointer

- The source to cause errors(too flexible)(segmentation fault…)



- int arr[4]:
- &arr[1] vs *(arr+1) vs &arr+1
- arr, arr+0, arr+1…
- *arr,*(arr+3)

C pointer

- int* arr[4]   vs   int (*arr)[8]
- int (*test)(int*)
- int* (*test)(int*)

- So the pointer is very flexible…
- You need to review c pointer carefully.

Linked list

- Define:
- typedef struct nodeT{
- int x…;
- struct node *next; } node;
- Create:
- struct node *root;
- root = (struct node *) malloc(sizeof(struct node));
- root->next = 0;
- root->x = 5;
- But usually the root is just a pointer, don’t contain informtion.

Linked list

- Add:
- struct node *new;
- new = (struct node *) malloc(sizeof(struct node)); (casting)
- new->x = ..;
- root->next = new;
- new->next = NULL;

Linked list

- Remove and search:
- While(tmp->next!=NULL)   Right?
- (1) the linked list is empty?
- (2)delete the first or last node?
- We talk about single linked list? What about two direction linked list? Leave for you. Have a good weekend…. Don’t worry, we will not confront this…..

Policy for grading

- Do make sure to book a slot for interview.(If not,0)
- If you book a slot, but neither come nor explain to me the reason.(0)
- If you explain me the reason and I think it is indeed a reason, I will give you a second chance. If not, let William handle you.
- One week late for submit. (-20%) Later, 0.
- 10% for the code style.
- Copy other's code, I will take you to William.

Policy for grading

- But I am indeed a very kind person.

- Last semester, even some students take advantage of my kindness…
- If you work hard, you will absolutely get 100.
- I will not find fault with everyone.

Office Hour

- I am not sure the time for that.
- Later I will post it on moodle.
- If you come, take your computer.
- I can help you , but I will not debug for you…
- If you have problems, feel free to email me.
- In the email, state your problem clearly, you can just take a screenshot.