Recitation 1 – linprog

Aug 28th

Silei Song

About me

- Silei Song
- Ph.D candidate at CS department
- email: song@cs.fsu.edu
- Office hour: Tu & Th 10:30 – 11:30, MCH 203D

linprog log in - Macbook

- open your terminal
- ssh [your cs id]@linprog.cs.fsu.edu
- enter your password

Transfer file

- open your terminal at the folder that contains the file
- scp [file name] [your cs id]@linprog.cs.fsu.edu:[route/targetfolder]
- enter your password

linprog log in - Windows

https://system.cs.fsu.edu/new/newuser/ssh-how-to/

- Download Tectia
- for your first connection (download & install may also happen), if your FSUCS id and password cannot work, try:
  - User Name: sshcs
  - Password: letmedownloadit

Transfer data

- Tectia file transfer GUI system

common linux command

- cd [folder name]: enter folder
  - return to higher level folder: cd ..
  - return to the top level folder: cd /
- ls: list content files
  - with details: ls –l
  - with specific extension: ls *.[extension]
- mkdir [new folder name]: make a new folder
- mv [file name] [new target position/new file name]: move file
- rm [file name]: remove file
  - remove whole folder: rm –rf [folder name]
- cp [file name] [target position]: copy file

vim

- vim [your file/new file]
- i – into insert mode
- v – into visual mode
  - using cursor to select
  - y – copy; d – cut; p/P - paste
- esc - break the current mode
- :q – quit file editor
- :wq – save and quit (:w is just save)
- :q! – quit without save

compile and run

- g++ -Wall –pedantic –o [exefile] [sourcecode.cpp]
- compile multiple files (header files)
  - generate .o files: g++ -c [file1.cpp] [file2.cpp]
  - generate executable files: g++ -o [exe file] [file1.o] [file2.o]
- run
- [exefile] [arguments] < [input] > [output]

- gdb [executable file] – to run file in the debug mode
  - q – to quit debug mode

gdb instructions

run or r	Executes the program from start to end.
break or b	Sets a breakpoint on a particular line.
disable	Disables a breakpoint
enable	Enables a disabled breakpoint.
next or n	Executes the next line of code without diving into functions.
step	Goes to the next instruction, diving into the function.
list or l	Displays the code.
print or p	Displays the value of a variable.
quit or q	Exits out of GDB.
clear	Clears all breakpoints.
continue	Continues normal execution

makefile

- to construct a makefile: vim Makefile (or makefile)
- assign variables: (when use, use $() to represent variables in codes)
  - CC = g++ # Compiler
  - CFLAG = -Wall –pedantic # Compiler flags
- Compile multiple .cpp files
  - SRCS = main.cpp print.cpp factorial.cpp multiply.cpp
  - %.o: %.cpp $(CC) $(CFLAG) -c $< -o $@ #for all .cpp files, compile to .o
  - OBJS = $(SRCS:.cpp=.o)
  - main: $(OBJS) $(CC) $(CFLAG) -o $(TARGET) $(OBJS)
- extra command: clean
  - clean: rm [file1 you want to delete] [file2] …
  - make clean
- make - default make Makefile
  - make –f [your make file]

tar

- Your simple Makefile (if no further instruction):

write your compiling instructions directly in Makefile

  - %.o: %.cpp

    g++ -Wall –pedantic -c [file1.cpp] [file2.cpp]

  - [exe file] : [file1.o] [file2.o]

    g++ -Wall –pedantic -o [exe file] [file1.o] [file2.o]

- tar file: tar -cf [archive.tar] [file1] [file2] [file3]
  - tar –cf [route/folder name]
- unzip tar file: tar -xf [archive.tar]
  - tar -xf [archive.tar] -C [specific location]