

# Programming assignment 1 walkthrough

Tao

# PA 1 walkthrough: Part I

- VirtualBox Install
- Ubuntu OS install
- Package install
  - Dependencies
- OS161, SYS161

# OS161, SYS161 ??

- SYS161: system architecture
- OS161: operating system
  
- Analogies:
  - SYS161: x86, x64(x86\_64), ARM, PowerPC
  - OS161: Windows, Linux, macOS, iOS
    - Kernel/User applications

# OS161, SYS161 tools

- binutils -> generate os161 executables
- gcc -> compile C to os161 assemblies
- gdb -> for you to debug C code compiled for os161
- sys161 -> a simulator of sys161 architecture
- os161 -> the actual source code of os161 operating system

# What have you done (Important facts!)

- On your (x86) machine, you built a compile/debug suite for os161 system in sys161 architecture
  - binutils, gcc, gdb
    - cs4300-ld,cs4300-gcc, cs4300-gdb
  - These are usually called cross-compiler. (Runs on x86, but generates executables for os161)
  - You can't directly execute the result from the cross-compiler
- You built a sys161 simulator that can specifically simulate run "sys161 executables" on your (x86) machine
- You built an *operating system* "os161" for sys161 architecture
- You execute os161 on your simulated sys161 environment (sys161)



# You built an operation system os161

- Kernel:
  - cs4300-os161/os161-1.11/kern
  - Runs on sys161 simulator
- User space utilities
  - cs4300-os161/os161-1.11/
  - Runs on os161 kernel
- Both are built by os161 compiler
  - (yes, os161 compiler can generate “bare” executable for sys161)

# Directories

- Downloads/ -> temporary source code
  - binutils-os161/
  - gcc-os161/
  - gdb-os161/
  - sys161/
- sys161/ -> your (cross-)compile suite, simulator
  - bin/ -> renamed (linked) compiler "cs4300-"s and the simulator "sys161"
  - tools/ -> original executables installed by source code
- cs4300-os161/os161-1.11/ -> your os directory
  - kern/ -> kernel part
- cs4300-os161/root/ -> the environment you can start a simulation

# Compilers

- gcc, gdb: your host compiler tools
  - Generate/debug linux-x86 executables
  - sys161 is built by your host compiler
- cs4300-gcc, cs4300-gdb: your cross-compiler tools
  - Generate/debug os161/sys161 executables
  - os161 kernel.
  - os161 userspace utilities.
  - If it complains “command not found”, check your \$PATH



## Part II: C programming

- `gcc hello.c -o hello`
- `./hello`
  
- File operations?

Questions?

# FAQ:

- VirtualBox no graphic -> Settings->graphics controller->VBoxVGA
- cs4300-gcc not found (or cse4300-)
  - Check \$HOME/sys161/bin directory (pg.7). Tools exist?
    - Yes -> \$PATH problem (“export” command and the red text in pg.3)
    - No -> redo “sh -c ...” command at correct directory (make sure it doesn’t show error)
  - Is prefix correct? Redo the ./configure using correct prefix.
- Other strange error messages during ‘make’:
  - Check and redo ‘./configure’ command exactly as shown in pdf
  - Differentiate ` and ‘.
- Error during ‘./configure’
  - Redo ‘tar ...’ step and change configure file content (egrep line).
- If you changed ./configure parameters, do a “make distclean”
  - If error still pops up, remove the directory and restart from decompressing