

Prelab 4. Compiling Programs (Due date: 4:00pm April 7, 2023)

- The TLCL book is publicly available at: <http://sourceforge.net/projects/linuxcommand/>
- Prelab report should be your individual work (a prelab report is not a team assignment). It aims to promote every student to read the textbook as the course is moving forward.
- HGU CSEE Standard on assignments:
 - Submitting assignments or program codes written by others or acquired from the internet without explicit approval of the professor is regarded as cheating.
 - Showing or lending one's own homework to other student is also considered cheating that disturbs fair evaluation and hinders the academic achievement of the other student.
 - It is regarded as cheating if two or more students conduct their homework together and submit it individually when the homework is not a group assignment.
- Submit your complete prelab report in PDF.

1. Read Chapter 23 of TLCL (pages 350-361) carefully. Write an answer to each of the following questions.

- a) Fill in the blanks.
- 1) The final result of the compilation process is the (executable program file).
 - 2) Compiling is the process of translating (source code) into the native language of the computer's processor.
 - 3) The availability of (source code) is the essential freedom that makes Linux possible.
 - 4) CPU executes programs in (machine language) that is a numeric code that describes extremely small operation.
 - 5) (Assembly language) replaces the numeric codes with slightly easier to use character mnemonics.
 - 6) (Compiler) converts programs written in high-level programming languages into machine language.
 - 7) The (configure) program is a shell script whose job is to analyze the build environment.

- b) Explain linker and libraries. What are they, and why are they necessary?

linker: used to form the connections between the output of the compiler and the libraries that the compiled program requires.

libraries: which contains multiple routines, each performing some common tasks that multiple programs can share.

There are many common tasks performed by programs. Take, for instance, opening a file. Many programs perform this task, but it would be wasteful to have each program implement its own routine to open files. It makes more sense to have a single piece of programming that knows how to open files and to allow all programs that need it to share it. Providing support for common tasks is accomplished by libraries. And linker connects the output of the compiler and the libraries that the compiled program requires.

- c) Explain portability in Linux.

Portability in Linux refers to the ability of source code to run on multiple Unix-like systems with minimal adjustments. This is important because different systems may have slight variations in their configuration or libraries, which can affect the way software is built and run. To make code portable, developers may use techniques like conditional compilation, which allows different code to be compiled depending on the system it's running on. Additionally, tools like configure are used to analyze the build environment and check for necessary external components. When running configure, it's important to prefix the command

with `"/"` to indicate that the program is located in the current working directory. Overall, portability is an important consideration for developers who want their code to run on multiple platforms without requiring major changes.

- d) What is the `/usr/local/bin` directory for in Linux?

the `/usr/local/bin` directory in Linux is a traditional location for locally-built software that has been installed using the `"install"` target in the makefile. It is intended to provide a separate location from system-provided directories and is typically not writable by ordinary users, which means that installations must be performed by the superuser or root.

- e) Read each statement and indicate whether it is true or false.

The makefile is a configuration file that instructs the make program exactly how to build the program.	<input checked="" type="checkbox"/> True / False
Even if makefile does not exist, the make program runs and takes a default compile sequence.	<input checked="" type="checkbox"/> True / False
Makefile describes the relationships and dependencies among the components that comprise the finished program.	<input checked="" type="checkbox"/> True / False
Each time the make program runs, the compile process starts from the scratch to prevent potential mistakes.	True / <input checked="" type="checkbox"/> False
One of the main features of make is to keep targets up-to-date by insisting that targets be newer than their dependencies.	<input checked="" type="checkbox"/> True / False

매번 compile 부터 시작하는 것이 아니라
목적파일이 있다면 link해주는
것부터 시작할 수도 있는 것이다