HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

OPERATING SYSTEM – LAB 02

Student name: LE HUY HOANG
Student ID: 1652212
Date: September 14th , 2018

Lecturer: PhD. Pham Hoang Anh

1. There are 4 compiling stages:
   - Pre-processing: do follows:
     + Removal of Comments
     + Expansion of Macros
     + Expansion of the included files.
   - Compiling: compile C program files into assembly code
   - Assembly: convert by assembler to machine code, each C file is converted to .o file
   - Linking: Linker links functions' prototype and its implementation

2. * transmit.c and transmit.h are compiled to transmit.o file
   * receive.c and receive.h are compiled to receive.o file
   * main.c is compiled to main.o file
   all *.o files are linked in Linking stage to 1 executable file

3. Because in the first time compiling, necessary files (source files) are loaded from disk to RAM and they are still be there in an interval of time even if OS has finished with them. As a result, later compiling, the compiling process takes place faster since they have been already in RAM.

4. Belows are in makefile used for Python:

```
HOST=127.0.0.1
TEST_PATH=./

clean-pyc:
    find . -name '*.pyc' -exec rm --force {} +
    find . -name '*.pyo' -exec rm --force {} +
   name '*~' -exec rm --force  {}
```

```
clean-build:
    rm --force --recursive build/
    rm --force --recursive dist/
    rm --force --recursive *.egg-info

isort:
    sh -c "isort --skip-glob=.tox --recursive . "

lint:
    flake8 --exclude=.tox

test: clean-pyc
    py.test --verbose --color=yes $(TEST_PATH)

run:
    python manage.py runserver

docker-run:
    docker build \
      --file=./Dockerfile \
      --tag=my_project ./
    docker run \
      --detach=false \
      --name=my_project \
      --publish=$(HOST):8080 \
      my_project
```

5. Assume the program consists of header file (foo.h) and source file (foo.c):

```
home
 |
 |-- headers
 |    |
 |    |-- foo.h
 |-- sources
 |    |
 |    |-- foo.c
 |-- main.c
```

and we are in "home"

makefile:

```
foo: foo.o main.o
	gcc foo.o main.o -o prog

./sources/foo.o: ./sources/foo.c ./headers/foo.h
	gcc -c ./sources/foo.c
main.o: main.c ./headers/foo.h
	gcc -c main.c
```