Laboratory 4 *make* Utility

It is very easy to compile a single C source. But, the complexity increases when a large number of programs need to be compiled as the programs must be made re-usable by other programs and other users to improve the efficiency.

UNIX contains a famous tool called **make** to resolve the difficulty. The command reads a file called **Makefile** by default, where all file dependencies are described. You can learn from the tutorials at http://www.ece.iupui.edu/tutorials/

For the project follow the instructions given below:

1. For a program who3.c on page 60 of the textbook, you must create C source files and header files for building the program "who3" to show the same output as a system program "who" generates.
2. Note that the program needs other functions on page 49, 58, 59, and 60.
3. You create two header files "show.h" and "utmplib.h". "show.h" contain definitions and function prototypes for show_info() and showtime(), and "utmplib.h" contains definitions and prototypes for each function on page 58-60.
4. You create C source file (.c) for each function. For instance, you create utmp_open.c file for the function utmp_open ( ).
5. Draw the dependency graph for the program who3.
6. You make a text file named "Makefile" conforming the following if the target is older than its depending files.
    a) Use "make libshow.a" for compiling functions defined in "show.h" and create a library named "libshow.a".
    b) Use "make libutmplib.a" for compiling function defined in "utmplib.h" and create a library named "libutmplib.a"
    c) Use "make lib" for creating both libraries.
    d) Use "make" to compile all your C files and build the executable "who3
    e) If you specify the target name for a function name (for example: make utmp_open), it will compile only the corresponding target and all the functions associated with it.
    f) If you give "clean" target, it will clean all intermediate files but not executable programs.
    g) If you give "cleanall" target, it will clean all intermediate files and executable programs.

    Note 1: Sun Solaris Unix and Redhat Linux uses a little different format of utmp called utmpx. Refer the man page for "who" and "utmpx" for more information. You will need to adjust the code to use the data structure utmpx instead of utmp. The calls are not difficult to see in the existing code.

    Note 2: You need to understand Unix commands "gcc", "ar", and "rm" for this lab.