Problem Statement for Senior Applicants to Unix Batch Dec-2016

Dear Senior Unix Applicant,

In order to gain admission to Unix Batch Dec. 2016 as a senior student, you've to fulfil following criteria.

[1] Attend first six lectures. This is compulsory.
[2] Read the problem statement below. Solve and submit it under 31st Dec. 2016.

Implementation of mini-gcc command

Consider that you've hello_world.c, C source file, containing simple C code to print "Hello, World!" on the console. 'hello_world.c' source file has to go through multiple translations before we get executable file corrossponding to it. The translation process is as follows.
.c ---Compilation---> .S -----Assembly----->.o------Linker------> executable.

Following commands show, how can you translate .c file into executable in step-wise manner (unlike gcc -o hello_world hello_world.c, which will hide intermediate steps).

- **Compiler command**[1]:
  ```
  # gcc -S -o hello_world.S hello_world.c
  ```
- **Assembler Command :**
  ```
  # as -o hello_world.o hello_world.S
  ```
- **Linker Command :**
  (For 32 bit Linux machine, following is the linker command)
  ```
  # ld -o hello_world -lc -dynamic-linker /lib/ld-linux.so.2
    hello_world -e main
  ```
  (For 64 bit machine, following is the linker command)
  ```
  # ld -o hello_world -lc -dynamic-linker
    /lib64/ld-linux-x86-64.so.2 hello_world -e main
  ```
  (Make sure that new DON'T enter new line while typing ld command)

Your task is to implement my_gcc.c program using "The Unix System Call Interface"[2] and standard C library functions, which will accept input .c file and

- will produce .S file if -S option is provided,
- will produce .o file (and delete .S file) if -c option is provided
- will produce executable file (and delete .S and .o files) if neither -S nor -c option is provided.
- If -o switch is provided to specify custom output file name then executable file will have custom name otherwise default 'a.out' name should be used.
- Additionally your program should generate 'build.log' file which will log all the entires corrossponding to file creations and file deletions.
  For example if command

`#./my_gcc -c -o hello_world.o hello_world.c`
is executed then build.log file will have entries similar to following.
Wed Dec 7 00:57:26 2016 : hello_world.S file is created.
Wed Dec 7 00:57:48 2016 : hello_world.o file is created.
Wed Dec 7 00:58:17 2016 : hello_world.S file is deleted.

- If you rebuild the program to produce either of assembly file, object file, or executable file then check whether older versions of the files are already present, (you can assume that assembly file corrossponding to hello.c source file will always have hello.S name, and likewise for object and assembly file). You will build source file ONLY IF last modified time stamp of source file is GREATER than last modified time stamp of target file (.S, .o or executable). You will make entries accordingly in build.log file.
  e.g. if you execute `# ./my_gcc -S -o hello.S hello.c` and assume that hello.S is already present in current directory and last modified time stamp of hello.S is greater than that of hello.c file then you'll log,
  Wed Dec 7 01:05:32 2016 : hello.S is upto date.
  And if last modified time stamp of hello.c is greater than that of hello.S file then you will delete hello.S file and recreate it, therefore, the entires will read,
  Wed Dec 7 01:07:18 2016 : hello.S is deleted.
  Wed Dec 7 01:07:43 2016 : hello.S is created.
- Lastly, make sure that if compilation fails, then you should abandon the subsequent build process if any.

Additional Notes :

- [2]Although assignment mandates usage of the Unix system call interface, because of unavailability of True Unix systems, you can use any POSIX compliant operating system viz. Linux, Solaris, FreeBSD etc. Make sure that you DO NOT use any system call which is not part of POSIX but is part of OS that you are using. You are free to use library functions provided by GNU C library which may not be part of ISO-C99 standard library, for example, getopt_long.
- Colour convention :
  - Text in read : Shell commands
  - Text in blue : Log entries in build.log file.
- In case you have not understood any part of problem statement or you've ambiguity regarding certain aspects of the output then contact Yogeshwar, either on
  - Mobile: +91 956 15 47 043 or on
  - Email : yogeshwar.private@gmail.com

Foot note :
[1] Strictly speaking, /usr/bin/gcc is not a compiler executable. Compiler executable name is cc1, but as it's path is not standard and varies from distribution to distribution, use gcc -S to compile .c to .S.