Assignment 2

Submission instructions:

Your submission should be a single tar file, preferably compressed, which contains:

• A single text file contains the script for question 1.

• A text file contains the answer for the question 2. (Hints: to fully understand the script, you are advised to type in the given script using vi editor, and make it executable and test it).

• A text file contains the answers for question 3.

• A workable makefile for question 4.

• A text file contains all the RCS commands you use for question 5.

Question 1. (10 marks)

Write a Bash script to delete comments from a C program. In Bash script, you may use sed commands. A C comment starts with the two-characters token (/*) and ends with the two-character token (*/). Assume that each comment line contains the start and end tokens. You need to pay attention to the following cases:

/* a comment line in a C program */

printf("/* print a comment line */\n");

x = 5; /* assign 5 to x */

[TAB][SPACE] /* another comment here */

Question 2. (15 marks)

A Bash script is given as follows.

```
#!/bin/bash
#A script demonstrating an until-loop and command line processing
#
# List the regular files of a directory greater than a given size

  name=${0##*/}
  Usage="Usage: $name [-h] [-s N] [directory]"

 if [ $# -eq 0 ]; then
 echo $Usage
 exit 1

fi

  until [ $# -eq 0 ]
do

    case $1 in
      -s) shift
          size=$1
          shift;;
     -h) echo $Usage
          exit 0;;
        *) directory=$1
           shift;;
     esac
done

directory=${directory:-‘pwd‘}

if [ ! -d $directory ]; then
      echo "‘$directory’ is not a directory"
      exit 1
fi
    arg=’’

if [ $size != ’’ ]; then
arg="-size +$size"

fi

find $directory $arg -type f -exec ls -l {} ’;’
```

Carefully look at it. Add comments to this script — line by line.

Question 3. (10 marks)

In the makefile below, identify:

• targets

• construction commands

• prerequisites

• macros

```
COBJECTS = menu.o users.o resellers.o propspects.o
FHILES = menu.h


leads: $(COBJECTS)
        gcc -o leads $(COBJECTS)

menu.o users.o resellers.o prospects.o: $(HFILES)
```

Question 4. (15 marks)

Write a makefile that reflects the following relationships:

1. The C source files transaction.c and reports.c are compiled to produce an executable accts.

2. transaction.c and reports.c include a header file accts.h

3. The header file accts.h is composed of two other header files: trans.h and reps.h.

Question 5. (10 marks)

Use a small text file to experiment with the RCS system. Note that RCS treats all ASCII file the same way; the files do not necessarily be programs, though usually are programs. Specially you should

• Place a Log keyword in the file

• Complete a set of check in operations which result in the RCS file structure depicted in Figure 1.

You will need to make small changes to the source file between check in operations.

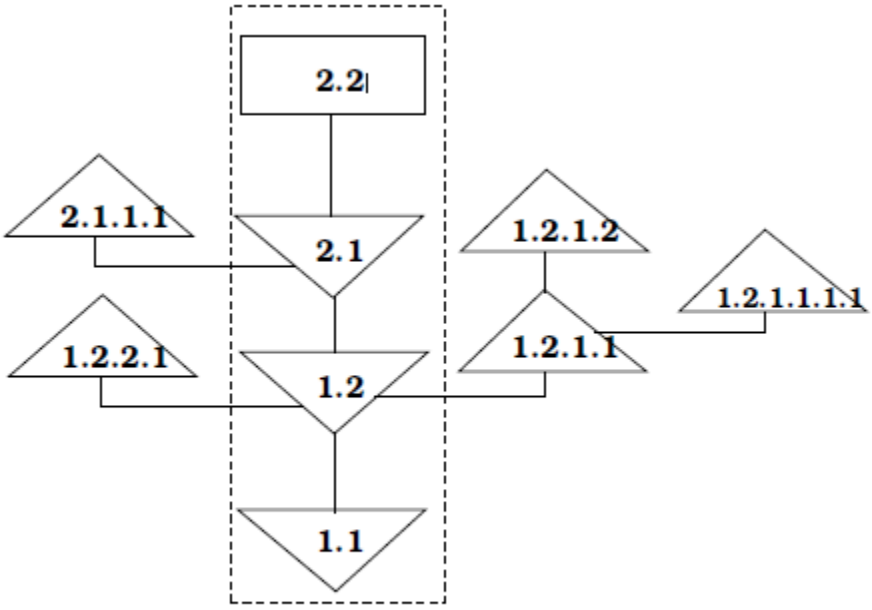

Figure 1:

• Associate a symbolic name with revision 2.1.1.1

• Make the branch containing revision 2.1.1.1 the default branch

• Use rlog to verify the RCS structure.