

CSI 402 – Systems Programming – Spring 2013

Programming Assignment II

Date given: Feb. 19, 2013

Due date: Mar. 1, 2013

Weightage: 5%

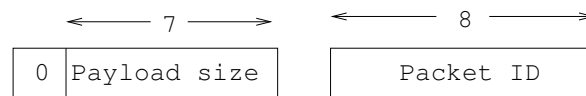
The regular deadline for this assignment is **11 PM, Friday, March 1, 2013**. With lateness penalty, the program will be accepted until **11 PM, Sunday, March 3, 2013**. The assignment will *not* be accepted after 11 PM on Sunday, March 3, 2013.

Very important: Your source program must consist of two or more C files and you must also have a **makefile**. (Additional information regarding this requirement is given later in this handout.) The C files (with extension “.c”), header files (with extension “.h”) and the **makefile** must be submitted together using the **turnin-csi402** command. Instructions for using **turnin-csi402** and additional specifications for the **makefile** will be included in the **README** file for this assignment.

The total grade for the assignment is 100 points, with 85 points for correctness and 15 points for structure and documentation.

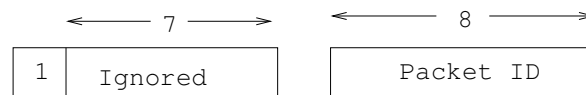
You will be writing a program that processes a binary (i.e., unformatted) file storing the packets collected during a network simulation. Assume that there are two different types of packets, namely **data packets** and **control packets**. The formats of the two types of packets are explained below.

Data packet format: Each data packet has a two byte header as shown below.



In the first byte of the header, the most significant bit for a data packet is 0. The remaining 7 bits of the byte specify as an *unsigned integer*, the payload size of the data packet; this is the number of bytes, *excluding* the header, included in the data packet. (Thus, the payload size can vary from 0 to 127 decimal.) The second byte of the header gives, again as an unsigned integer, the ID of the data packet. (Thus, the ID of a data packet can vary from 0 to 255 decimal.) For any data packet, the two bytes of the header are followed by a number of bytes equal to the payload size. (Of course, if the payload size is zero, then the data packet consists of just the two bytes forming the header.) The contents of the payload bytes are not relevant for this program.

Control packet format: A control packet consists of just a two byte header as shown below.



In the first byte of the header, the most significant bit for a data packet is 1. (This is how we can distinguish between a data packet and a control packet.) The remaining 7 bits of the byte are ignored. The second byte of the header gives, again as an unsigned integer, the ID of the control packet. There is no payload for a control packet.

Note that the input file may have data and control packets interspersed. Also, the data and control packets may *not* be in sorted order of IDs. Suppose the input file contains control packets with IDs 0, 5, 6, 9 and 14. Then, the missing control packets are assumed to be those with IDs 1,

2, 3, 4, 7, 8, 10, 11, 12 and 13. (Thus, we don't worry about control packets whose ID exceeds the largest control packet ID found in the file.) A similar comment applies to data packets.

The executable version of your program for Part (a) must be named **p2**. It will be executed by a command line of the following form:

p2 *flag* *infile*

The command line parameter *infile* represents the name of the (unformatted) input file containing data and control packets. The parameter *flag* may be one of the strings "-c", "-d" or "-cd", with the following significance.

- (1) If the flag is "-c", your program must produce to **stdout**, the following information about control packets: the number of control packets in the file, the largest control packet ID and the IDs of missing control packets. In particular, missing control packet IDs must be output as shown in the following example.

Example: Suppose the input file contains control packets with IDs 0, 5, 6, 9 and 14. In this case, the number of control packets in the file is 5 and the largest control packet ID is 14. Missing control packet IDs (as explained above) are 1, 2, 3, 4, 7, 8, 10, 11, 12 and 13. The list of missing control packet IDs must be output as follows:

Missing control packet IDs: 1-4, 7, 8, 10-13

In particular, note that any sequence of *three or more* consecutive missing IDs is expressed as a range.

- (2) If the flag is "-d", your program must produce to **stdout**, the following information about data packets in the file: the number of data packets, the largest data packet ID, the largest payload value and the IDs of missing data packets. The missing data packet IDs must be output in the same fashion as that for control packets.
- (3) The flag is "-cd" is equivalent to specifying both "-c" and "-d". Thus, for the "-cd" flag, your program should output to **stdout**, the above mentioned information for both control and data packets.

Your program must detect the following command line errors. You may assume that there are no errors in the input file.

- (1) The number of command line arguments is not equal to three.
- (2) The specified input file specified can't be opened.
- (3) The flag specified on the command line is not one of "-c", "-d" or "-cd".

In each of the above cases, your program should produce a suitable error message to **stderr** and stop.

Structural requirement: Your submission must have *at least two* C source files, zero or more header files and a **makefile**.

Information about README file: The README file for this assignment will be available by 10 PM on Saturday, February 23, 2013. The name of the file will be **prog2.README** and it will be in the directory **~csi402/public/prog2** on **itsunix.albany.edu**.