CS 286 Computer Organization & Architecture  
Fall 2023  
Programming Project #3

Submission Due: 12:00 p.m. (noon) on December 4, 2021

Assignment Descriptions:

Each computer connected to the Internet is assigned a unique ID, called “IP address”. In this assignment, we will develop an IP-address analyzer using MIPS R-3000 assembly instruction set. Since a unique IP address is assigned to each host computer in the Internet, other computers identify and locate each host computer in the Internet using IP addresses. In the Internet, each individual computer belongs to a group of computers, called “(network) domain”. For example, SIUE is a domain in the Internet, and every computer in SIUE campus belongs to SIUE domain. To reflect this design, each IP address consists of two internal address components of “domain address” and “host address”. The domain addresses represent a network domain, while the host addresses represent the address of each individual computer in a domain. Regarding the domain address, the Internet assumes three different domains, known as class-A, -B, and -C domains. Class-D domains are special domains, called “multicast domains” (Class-D domains are not covered (i.e., ignored) by Project #3).

The three different domain classes represent domains in different size (number of host computers a domain can have). For example, each class-A domain can accommodate up to $2^{24}$ host computers, while a class-B domain for up to $2^{16}$ computers. A class-C domain can accommodate up to $2^8$ computers. Each IP address is represented as a combination of four positive integers, where each integer can take 0 ~ 255 (as an eight-bit unsigned integer). This means that each IP address consists of a total of 32 bits (4 bytes). The 32 bits in each IP address are split to the domain and the host address (see the figure below). For example, since each class-A domain uses 24 bits to represents each host computer, the first 8 (=32 -24) bits are used to represent their domain addresses. Therefore, as many as $2^{(32-24)} = 2^8$ (= 128) class-A domains can exist in the Internet. Similarly, there can be $2^{(32-16)} = 65,536$ class-B domains and $2^{(32-8)} = 2^{24}$ class-C domains in the Internet.

![IP Address Diagram](image-url)
The four domain classes of IP addresses are classified using the first number in IP addresses as follows.

<table>
<thead>
<tr>
<th>Domain Class</th>
<th>First Number Range (Decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1-127</td>
</tr>
<tr>
<td>B</td>
<td>128-191</td>
</tr>
<tr>
<td>C</td>
<td>192-223</td>
</tr>
<tr>
<td>D</td>
<td>224-239</td>
</tr>
</tbody>
</table>

Hardware equipment, called “routers”, are responsible for forwarding network traffic to a correct direction in the Internet and they use the domain addresses to do so (i.e., the Internet routers perform routing using only the domain addresses).

Assignment Requirements:

Requirement #1 (25/100 points): Your program (using MIPS R-3000 assembly instructions) analyzes a given IP address for its domain class and displays the domain class the given IP address belongs to. Your program should prompt for an IP address and display its domain class (as shown below).

Any invalid input should be detected as soon as a number is entered and your program should repeat a prompt until a valid number is entered. See the figure below.

Once a valid number is entered, any such valid input should not be repeated (as show below).
After all the four numbers are correctly input, your program should determine its domain class and display it.

Requirement #2 (75/100 points): Your program scans an IP routing table for a matching line in the table in terms of the domain addresses (the host addresses should be ignored). Your program should satisfy the following requirements:

(1) Your program should find “the line number (the first number in an IP table)” for the last match. If more than one line matches, your program should display the last match.

(2) If there is no match, your program should show that there is no match after all the lines in a routing table are scanned.

The following two figures show a successful and an unsuccessful search results.
Each routing table is defined as follows using PC-SPIM directive. The size of an IP routing table is defined as one “word (.word)” unsigned integer, which is associated with a label “IP_ROUTING_TABLE_SIZE:”. The beginning of the routing table is defined using the label “IP_ROUTING_TABLE”. Each of the four unsigned number in an IP address is declared using “.byte” directive, which indicates one byte (8 bits) in memory. The following sample IP routing will be posted to CS286 course home and it is suggested for each of you to use (you can modify it as you like) it for your testing and debugging. When the course TA grades your program, other IP routing tables (but in the same format) will be used for testing (the IP routing table at the beginning of your *.asm program will be replaced by another IP routing table). Each of you should copy the whole IP routing table (include “IP_ROUTING_TABLE_SIZE:")) at the beginning of your *.asm source code file (in its “data section”).

```
IP_ROUTING_TABLE_SIZE:
    .word 10

IP_ROUTING_TABLE:
    # line #, x.x.x.x .................................
    .byte 0, 146, 92, 255, 255  # 146.92.255.255
    .byte 1, 147, 163, 255, 255  # 147.163.255.255
    .byte 2, 201, 88, 88, 90  # 201.88.88.90
    .byte 3, 182, 151, 44, 56  # 182.151.44.56
    .byte 4, 24, 125, 100, 100  # 24.125.100.100
    .byte 5, 146, 163, 140, 80  # 146.163.170.80
    .byte 6, 146, 163, 147, 80  # 146.163.147.80
    .byte 10, 201, 88, 102, 80  # 201.88.102.1
    .byte 11, 148, 163, 170, 80  # 146.163.170.80
    .byte 12, 193, 77, 77, 10  # 193.77.77.10
```

Note: the contents of the routing table will be changed for testing (the table can have more than 10 IP addresses (the table size specified at IP_ROUTING_TABLE_SIZE will be changed (increased or decreased).
**Requirement #3:** The following is the definition of each IP routing table.

- IP Routing Table Size ("IP_ROUTING_TABLE_SIZE"): as one “word” unsigned integer (1 ~ 399), which indicates the number of IP addresses contained in the IP table.
- Line Number (no label associated to it): as one “byte” unsigned integer (1 ~ 255).
- The first number in an IP address: as one “byte” unsigned integer (0 ~ 255).
- The second number in an IP address: as one “byte” unsigned integer (0 ~ 255).
- The third number in an IP address: as one “byte” unsigned integer (0 ~ 255).
- The fourth number in an IP address: as one “byte” unsigned integer (0 ~ 255).

A set of the five numbers (② to ⑥) will be repeated for as many times as indicated by “IP Routing Table Size”.

**Requirement #4:** If an invalid number appears in an IP address (in an IP routing table), your program should detect it and should display each detected invalid number. For example if an IP routing table contains the following IP address:

192.351.0.99

The second number (‘351’) is invalid, since each number (except the line #) should be 0 ~ 255. When an invalid number is detected, your program should display it and should skip that IP address (one that contains an invalid number):

“Invalid number 351 is detected at line 7. The IP address is skipped.”

**Requirement #5:** It is your responsibility to develop a program that works for IP routing tables in the given format (as described in “Requirement #3” above).

**Requirement #5:** This project is an individual project. No idea sharing, partial program source-code sharing” or whole program source-code sharing is allowed. The TA will check the contents in each *.asm source code file.

**Expectation #1:** If multiple submissions are made by an individual, the last submission will be considered “the final submission” and the TA will grade only the final submission.

**Expectation #2:** Within 48 hours from the submission deadline, no new question about this project will be answered.

**Expectation #3:** There may be a question about Project # in the final exam, which asks fundamental concepts covered by Project #3. Each Project #3 grade can be adjusted if fundamental concepts covered by Project #3 is not adequately demonstrated through the final exam question(s).

**Suggestion:** When you are submitting your *.asm file by email(s), it is strongly suggested for each of you to “cc” your submission to yourself (and hfujino@siue.edu). After you receive your “cc-ed” submission, it is also suggested that each of you to test-run it (to make sure if it is the file you are finally submitting).
Submissions:

- Your source code file (only one *.asm file) should be submitted to the CS286 Moodle or should be e-mailed to the TA (gwitske@siue.edu) by 12:00 p.m. (noon) on December 4th.

- For those who stop by Dr. Fujinoki’s office by 48 hours prior to the submission due, immediate feedback will be provided (please make sure to bring your *.asm source code file).

Note: In your program source code, include your name, student ID (the last 3 digits) in your program header.

Late Submission:

- Penalty of -10% will be given for every 12 hours after the due (i.e., -10% for a submission within the first 12 hours after the submission due).

- Submission more than 24 hours after the due will not be accepted.

Grading Criteria:

- **Assembler error:** If the TA can not assemble your assembler source code file due to any assembler error: 0 ~ 20% of the credit for this assignment phase will be given.

- **Routing-table error:** Any error when the TA replaces the UP routing table (no matter if it is minor or major): 10~30% of the credit for this assignment phase will be given.

- **Run-time error:** After your assembly source code file is successfully assembled to machine codes, if your program does not satisfy any requirement: 5% penalty for each minor problem. For any major failure, penalty depends on each such major problem.