# General Information

|  |  |
| --- | --- |
| **Name:** |  |
| **Emp#:** |  |
| **Date/Time of Assessment:** |  |
| **Guidelines:**   1. Throughout this assessment, please make sure to follow **good coding guidelines** including proper indention, variable and function naming conventions etc. 2. Wherever required, please add simple aspects like **pseudo-code or block/ interaction diagram, flow chart** etc. 3. DO NOT FORGET to return this FORM back to the coordinator. 4. Attach all papers that were used in completing the assignment(s). 5. All questions are mandatory. They carry varying marks. 6. Your answer sheet will be used for the assessment process and therefore ensure legible handwriting. | |

# C Programming Skills

Provide answers to all the questions in this section.

**NOTE**: Most/All of the questions are interdependent and therefore pay attention to the dependence.

1. An IPv4 IP address consists of 32 bits of information. Define a data structure that can store a given IP address (so that it can be used for standard networking function calls).
2. A routing table is an internal data structure that contains fields as depicted in the below table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Network Destination** | **Netmask** | **Gateway** | **Interface** | **Metric** |
| 0.0.0.0 | 0.0.0.0 | 192.168.0.1 | 192.168.0.100 | 10 |
| 127.0.0.0 | 255.0.0.0 | 127.0.0.1 | 127.0.0.1 | 1 |

Define a C data structure to create and maintain 10 entries (rows) of the above table.

1. Implement C routines that perform the following functionality for the data structure defined in above:
   1. Initialize the table
   2. Add an entry to the table
   3. Invalidate/free an entry in the table
   4. Query fields in the table (for example what is the metric for a given Network Destination)
2. In case the data structure defined in above has to accommodate an unlimited set of entries (more than 10 entries) what are the main changes required in your implementations.

1. What are the C language constructs available to speed up the execution time of the functionality of Querying fields (refer point d. in question above).
2. Write a C program to implement the basic functionality of 'grep' command (to recursively search for a string through directories and sub-directories).

# Linux/OS Fundamentals

Provide answers to all the questions in this section.

1. What is a daemon process? What are the characteristics that need to be changed to make a process as a daemon process? List a few system daemons.
2. What is a shared library? How to create a shared library in a Linux system?
3. How can a parent process check if child is still running? How can this be checked without blocking?
4. What is a re-entrant function? Provide example code block for re-entrant code.
5. What are blocking and non-blocking IO operations?
6. What are the differences between fork, vfork and clone?

**-- End of Assessment --**