

# CSC 3550 Written Examination

## Exam Instructions:

You must use 5 different functional, OO, or scripting languages to complete this exam. Each member of the group must complete one of the questions individually. Two of the questions can be completed as a group effort.

All scripts/programs should read from stdin and print to stdout.

1. Write a script/program that parses the web log located at [http://users.csc.tntech.edu/~elbrown/access\\_log.bz2](http://users.csc.tntech.edu/~elbrown/access_log.bz2) (an older actual, readable log file from users.csc.tntech.edu). The input file should be read from the URL and not from a locally stored copy of the file. The results, that will be printed, to the screen are:
  - a. The top 20 addresses that have visited our pages (name and IP if possible)
  - b. A unique list of web browsers used to view our site listed in order of decreasing popularity paired with the operating system (i.e. Firefox – Windows, Safari – Mac OS X, Firefox – Linux, MSIE 5.0 – Windows)
  - c. The top 20 web pages accessed at our site (non 404 status)
  - d. Overall percentage of on-campus hits (anything from 149.149.x.x) verses off-campus hits
  - e. Average number of hits per day for each day represented in the file
  - f. Number of 404 and 500 status codes
2. Write a script/program that performs a system health check. This check will evaluate the health of core system services such as DNS, Active Directory/LDAP, time synchronization, file system status, file-sharing status. This script/program should run on both the Linux and Windows platforms.
3. Write a script/program that displays a “complete” profile of a user. Elements to be displayed would include but not be limited to last login time, total disk usage displayed in “human-readable” format, full name, and group memberships. This script/program should be run from your primary LDAP server (a.k.a. your Active Directory box). Provide optional flags that will display the last login date or disk usage result in ascending or descending order.
4. Write a script/program that parses the security log file for your system (Linux or Windows). This script should assist you in tracking anomalies such as an extreme number of failed login attempts from a specific user or from a specific site.
5. Write a script/program that archives the “important” configuration files on your system. Archiving the file would entail saving a copy of the file to another directory and creating a MD5 hash to determine if changes has been made to the original. The script/program will run in three modes {archive, check/update, restore}.