**CSE-381: Systems 2**

**Homework01**

**Due: Sunday, September 24, 2023 before 11:59 PM**

**Email-based help Cutoff: 11:59 PM on Friday September 22, 2023**

Max Points: 25

## **Objective**

Developing the C++ program from this homework assignment will help you:

* Understand the use of uid and gid in Linux (input data files are in the same format as the /etc/passwd and /etc/group files on a Linux system)
* Gain familiarity with developing, debugging, and testing of C++ programs
* Review concepts of and file processing.
* Review basics of problem solving
* Review the use of std::unordered\_map

## **Grading information**

* The program submitted for this homework must pass the first two base case tests in order to qualify for earning any score at all. Programs that do not meet base case requirements cannot be submitted.
* You will not be able to submit programs that do not compile, compile with warning messages, or which do not pass cpplint style checking. Apply the iterative development approach covered in class session 6 to ensure you have something to submit.
* **Do not use global variables** – that is not good programming practice.
* Base case is worth 12 points
* Passing additional tests is worth 8 points
* Formatting & Documentation is worth 5 points: If your comments are poor, you may lose points in this category

## **Project overview: Develop program to print group membership**

#### **Objective**

The objective of this program is to print the login IDs of the users belonging to a given set of gids (group IDs) specified as command-line arguments. The necessary data is read from two provided text files, passwd and groups.

#### **Background**

In Linux, users are internally represented using a unique number called user ID or uid. Moreover, a set of users can be logically organized into a *group*. Such groups are represented by

a group ID or gid. Typically, these numbers are rarely used. Rather, a **name** is associated

with these numbers and the names are referred to. This program will serve as an excellent tool to quickly identify membership in a given group.

#### **Data file formats**

Prior to solving any problem it is important to study the supplied data. So, make sure you view the data files: You can use more, less, or a Linux text editor like emacs, vim or nano for this. The supplied data files are in the same format as they are in a real Linux OS as described below. You will need to download these files to your virtual machine in order to read/use them.

* User data ([passwd](https://www.geeksforgeeks.org/understanding-the-etc-passwd-file/)): The supplied file contains user information in the following colon (:) delimited format:

loginID:passkey:uid:…

For example, the following line from passwd

amf:x:1000:1000:Alan Ferrenberg:/home/amf:/bin/bash

contains the login ID amf as the first entry, x is some passkey (not used) followed by the uid (an int) and the gid (an int). The rest of the information on each line is not used in this project.

* Group information ([group](https://www.thegeekdiary.com/etcgroup-file-explained/)): The supplied file contains group information in the following colon (:) delimited format:

groupID:passkey:gid:members…

For example, the following line from group,

adm:x:4:syslog,amf

contains the group ID adm as the first entry, x is some passkey (not used) followed by the gid (an int), followed by a comma separated list of uids. In the program, giving group 4 as the input would result in this output:

4 = adm: syslog(104) amf(1000)

where the uid for each loginID is shown in parentheses. The loginID for a uid is found in the passwd file, described just above.

#### **Sample outputs**

One you have completed your program you can test its operation using the command shown below and you can compare your output to the output also shown below. Note that group IDs are specified as command-line arguments. In Visual Studio Code, you can run the program at the command line to provide the command line arguments, or investigate on your own how you provide command line arguments to a program in VS Code. Note that this will require you to create/edit config files.

* Base case #1 [Must pass to earn any points]:

Test with exactly one valid group ID as a command-line argument, for a group with no members

$ ./MUid\_homework01 0

0 = root:

* Base case #2 [Must pass to earn any points]:

Test with exactly one valid group ID as a command-line argument, for a group with one member

$ ./MUid\_homework01 24

24 = cdrom: amf(1000)

* Base case #3 [Must pass to earn any points]:

Test with exactly one valid group ID as a command-line argument returning multiple uids

$ ./MUid\_homework01 4

4 = adm: syslog(104) amf(1000)

* Base case #4 [Additional feature]:

Test with an invalid group id

$ ./MUid\_homework01 999

999 = Group not found.

* Base case #5 [Additional feature]:

Test with many valid/invalid group IDs supplied as command-line arguments. You can check what the output for this should be by running the program for the individual gids

## **Notes/Tips**

1. Remember that you can use the CODE plugin multiple times to make sure your submission is valid and passes the required and optional test cases.
2. **DO NOT** use global variables!
3. Use std::ifstream to read data from the text files.
4. Use std::istringstream to process each line read using the std::getline method. See examples in Part 4 of the “Rapid overview of C++” videos.
5. Use an unordered\_map to store uid/loginID mapping information to ease look-up when processing group membership.
6. It would be easier to compute and store the line of output to print for each gid in another unordered\_map.
7. Here are useful #includes for your reference:

#include <iostream>

#include <string>

#include <fstream>

#include <sstream>

#include <unordered\_map>

#include <algorithm>

## **Submission Instructions**

This homework assignment must be turned-in electronically via Canvas CODE plug-in. Make sure of the following:

* Your C++ source code is named MUid\_homework01.cpp (where MUid is your Miami Unique ID, e.g., ferrenam\_homework01.cpp).
* Your program compiles successfully, without any warnings or style errors.
* You have documented the methods.
* You have tested the operations of your program as indicated and your code passes at least the first three base cases.

Once you have completed the assignment, upload just your MUid\_homework01.cpp source file to Canvas via the CODE plug-in as you’ve done for previous assignments.