Introduction

In this assignment, you'll write a program that will get you familiar with reading and writing files and directories on Unix.

Learning Outcomes

Describe the API for different operations related to files (Module 3 MLO 3)
Describe the API for different operations related to directories (Module 3 MLO 6)
What are scopes and types of permissions associated with files and directories? (Module 3 MLO 7)
Instructions

Write a program that

Reads directory entries
Finds a file in the current directory based on user specified criteria
Reads and processes the data in the chosen file
Creates a directory
Creates new files in the newly created directory and writes processed data to these files
Format of the CSV File

Here is a sample file whose format corresponds to the format of the CSV file your program will be tested with (this is exactly the same file format as in Assignment 1).

The first row in the file contains the column headers, and not movie data.
All other rows have the same format and no columns are missing in any row.
Commas appear as delimiters between columns, but will not appear in the values of any columns.
This file has the following columns:

Title
This is a string with the movie title.
E.g., Iron Man 2
Year
This is a 4 digit integer value for the year the movie was released
E.g., 2010
Languages
The language or languages in which the movie was released.
One or more string values that are always enclosed within []
Multiple values are separated by semi-colons.
E.g,
[English;Portuguese;Spanish]

[English;French]
[English]
You can assume that the maximum number of languages any movie can be released in is 5.
You can assume that the maximum length of a language string is 20 characters.
Rating Value
A number between 1 and 10 (inclusive of both 1 and 10)
It can be an integer or a double with one digit after the decimal point
E.g.,
5
8.7
Program Functionality

Main Menu

The program starts and presents two choices to the user

1. Select file to process
2. Exit the program
Enter a choice 1 or 2:
1. Select file to process

If the user picks this option, they are presented with 3 further choices about which file to process (see details below)
2. Exit the program

If the user chooses this option, the program should exit.
Notes:

You can assume that when the program asks user to enter an integer, the user will indeed enter an integer (i.e., you don't need to verify the data type of the user input).
For the interaction choice if the user enters an incorrect integer (i.e., something other than 1 or 2), print an error message and again present the 2 choices to the user.
Selecting a File to Process

If the user picks this option, they are presented with the following menu options

Which file you want to process?
Enter 1 to pick the largest file
Enter 2 to pick the smallest file
Enter 3 to specify the name of a file
Enter a choice from 1 to 3:
If the user picks 1

The program finds the largest file with the extension csv in the current directory whose name starts with the prefix movies_ and automatically process it.
In case of tie, pick any of the files with the extension csv starting with movies_ that have the largest size.
If the user picks 2

The program finds the smallest file with the extension csv in the current directory whose name starts with the prefix movies_ and automatically process it.
In case of tie, pick any of the files with the extension csv starting with movies_ that have the smallest size.
If the user picks 3

The program asks the user to enter the name of a file.
The program checks if this file exists in the current directory. If the file is not found, the program should write an error message and again give the user the 3 choices about picking a file, i.e., don't go back to the main menu, but stay at the menu for picking a file.
For this option, there is no requirement that the file name must start with a particular prefix or that it must have a particular extension.
After a file has been successfully picked based on any of the 3 options picked by the user, the program must print the name of the file that will now be processed.

For example, if the user picks the option for the smallest file, and the name of that file is movies_2.csv, then the following message will be displayed

Now processing the chosen file named movies_2.csv
Note: your program must print the message with the name of the chosen file. The items related to file choice will be graded based on this message.

Processing the File

The program now reads the chosen file and processes it. After processing the file, the program goes back to the main menu.

The goal is that whenever the program processes a file, it will

Create a new directory and print a message with the name of the directory that has been created
The directory must be named your_onid.movies.random
where
random is a random number between 0 and 99999 (both numbers inclusive)
your_onid is your ONID
E.g., when chaudhrn runs his program and processes two files the following 2 directories may be created (of course the random numbers can be different)

chaudhrn.movies.83465
chaudhrn.movies.25
The permissions of the directory must be set to rwxr-x---
i.e., the owner has read, write and execute permissions, and group has read and execute permission to the directory.
Parse data in the chosen file to find out the movies released in each year
In the new directory, create one file for each year in which at least one movie was released
The permissions on these files must be set to rw-r-----
i.e., the owner can read and write to the file, while group can only read the file.
The file must be named YYYY.txt where YYYY is the 4 digit integer value for the year.
E.g., the file for movies released in 2018 must be named 2018.txt
Within the file for a year, write the titles of all the movies released in that year, one on each line
E.g., if two movies Avengers: Infinity War and Mary Queen of Scots where released in 2018, then the file 2018.txt will have two lines with each of the two titles on one line each.
Note: The bullet points above state what the processing accomplishes. There is no requirement that your program does the processing in the order of these bulleted points.

Sample Program Execution

Here is a complete example of executing the program in a directory where

The file foo_bar does not exist
The file great_movies.csv exists
Of all the files with the extension csv and the prefix movies_
the largest file is named movies_1.csv
the smallest file is named movies_2.csv

```
1. Select file to process
2. Exit the program

Enter a choice 1 or 2: 1

Which file you want to process?
Enter 1 to pick the largest file
Enter 2 to pick the smallest file
Enter 3 to specify the name of a file

Enter a choice from 1 to 3: 1
Now processing the chosen file named movies_1.csv
Created directory with name chaudhrn.movies.89383

1. Select file to process
2. Exit the program
```

```
Enter a choice 1 or 2: 1

Which file you want to process?
Enter 1 to pick the largest file
Enter 2 to pick the smallest file
Enter 3 to specify the name of a file

Enter a choice from 1 to 3: 2
Now processing the chosen file named movies_2.csv
Created directory with name chaudhrn.movies.30886

1. Select file to process
2. Exit the program

Enter a choice 1 or 2: 1

Which file you want to process?
Enter 1 to pick the largest file
Enter 2 to pick the smallest file
Enter 3 to specify the name of a file

Enter a choice from 1 to 3: 3
Enter the complete file name: foo_bar
The file foo_bar was not found. Try again

Which file you want to process?
Enter 1 to pick the largest file
Enter 2 to pick the smallest file
Enter 3 to specify the name of a file

Enter a choice from 1 to 3: 3
Enter the complete file name: great_movies.csv
Now processing the chosen file named great_movies.csv
Created directory with name chaudhrn.movies.92777

1. Select file to process
2. Exit the program

Enter a choice 1 or 2: 2
```