GEBZE TECHNICAL UNIVERSITY
COMPUTER ENGINEERING FACULTY

CSE 344 SYSTEM PROGRAMMING

HOMEWORK 1 REPORT

ENES AYSU
1901042671

Program Description

The provided program is a command-line utility called "gtuStudentGrades" designed to manage student grades. It allows users to perform various operations such as creating a grades file, adding student grades, searching for a student's grade, sorting student grades, and displaying student grades in different formats. The program also logs each command executed into a log file named "log.txt".

Function and Processes

writeLogEntry(const char task):

- This function is responsible for writing a log entry to the "log.txt" file.
- It forks a child process to perform the logging operation asynchronously.
- Within the child process, it opens the log file, retrieves the current time, formats the log entry, writes it to the log file, and closes the file.

displayUsage():

- Displays usage instructions for the program's commands.
- Similar to writeLogEntry(), it forks a child process to execute the usage instructions asynchronously.
- It prints out the available commands along with their descriptions.

createGradesFile():

- Creates a grades file named "grades.txt" if it doesn't already exist.
- It forks a child process to create the file asynchronously.
- Checks if the file already exists or not and informs the user accordingly.

addStudentGrade(char *name, char grade, char filename):

- Adds a new student's name and grade to the specified grades file.
- Takes the student's name, grade, and filename as arguments.
- It forks a child process to perform the addition asynchronously.
- Opens the file for appending, constructs the student information string, writes it to the file, and closes the file.

readline(int fd, char buffer, size_t maxlen):

- Reads from a file descriptor until a newline character or EOF is encountered.
- Helps in reading lines from files efficiently.

searchStudent(char name, char filename):

- Searches for a student's grade in the specified grades file.
- Takes the student's name and the filename as arguments.
- It forks a child process to perform the search asynchronously.
- Opens the file for reading, reads each line, compares student names, and prints the grade if found.

sortAll(char filename):

- Sorts all student grades in the specified grades file based on user-defined criteria.
- It forks a child process to perform sorting asynchronously.
- Reads the file, stores student data in an array of structures, prompts the user for sorting criteria, sorts the data accordingly, writes sorted data back to the file, and prints sorted entries.

showAll(char filename):

- Displays all student grades from the specified grades file.
- It forks a child process to display grades asynchronously.
- Opens the file for reading, reads each line, and prints it.

listGrades(char filename):

- Displays the first five student grades from the specified grades file.
- It forks a child process to list grades asynchronously.
- Opens the file for reading, reads the first five lines, and prints them.

listSome(int numOfEntries, int pageNumber, char filename):

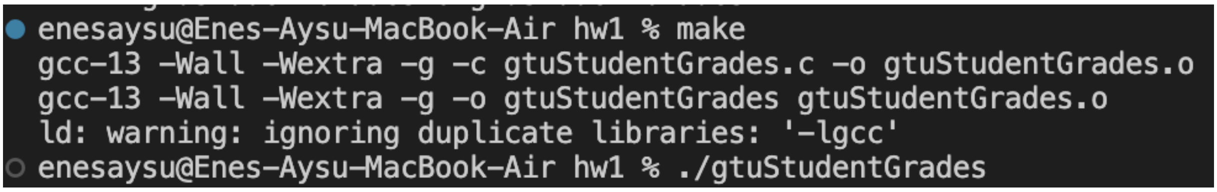
- Displays student grades based on the given page number and number of entries per page.
- It forks a child process to list grades asynchronously.
- Calculates the starting line number based on the page number and number of entries per page, skips lines accordingly, reads and prints the specified number of lines.

main():

- The main function of the program.
- Continuously prompts the user for commands until "exit" is entered.
- Parses user input executes the corresponding function for each command, and logs the command into the log file.

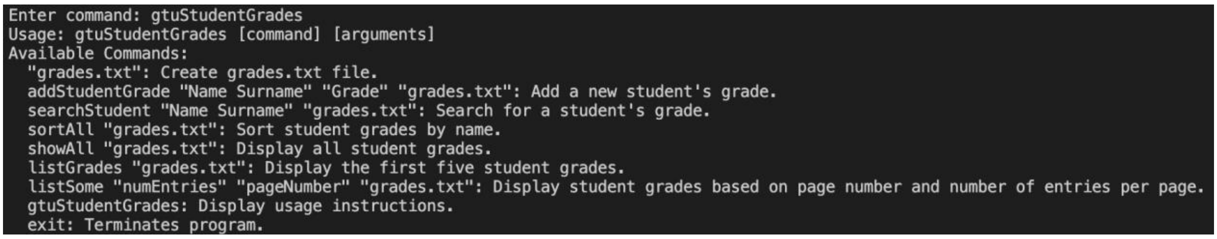
Tests

For Compilation



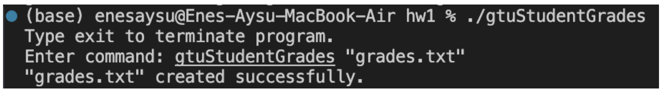
```
● enesaysu@Enes-Aysu-MacBook-Air hw1 % make
  gcc-13 -Wall -Wextra -g -c gtuStudentGrades.c -o gtuStudentGrades.o
  gcc-13 -Wall -Wextra -g -o gtuStudentGrades gtuStudentGrades.o
  ld: warning: ignoring duplicate libraries: '-lgcc'
○ enesaysu@Enes-Aysu-MacBook-Air hw1 % ./gtuStudentGrades
```

gtuStudentGrades



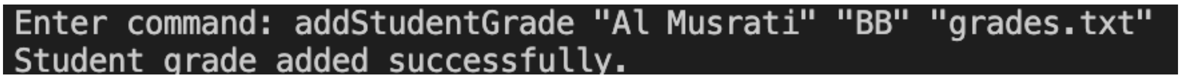
```
Enter command: gtuStudentGrades
Usage: gtuStudentGrades [command] [arguments]
Available Commands:
  "grades.txt": Create grades.txt file.
  addStudentGrade "Name Surname" "Grade" "grades.txt": Add a new student's grade.
  searchStudent "Name Surname" "grades.txt": Search for a student's grade.
  sortAll "grades.txt": Sort student grades by name.
  showAll "grades.txt": Display all student grades.
  listGrades "grades.txt": Display the first five student grades.
  listSome "numEntries" "pageNumber" "grades.txt": Display student grades based on page number and number of entries per page.
  gtuStudentGrades: Display usage instructions.
  exit: Terminates program.
```

gtuStudentGrades “grades.txt”

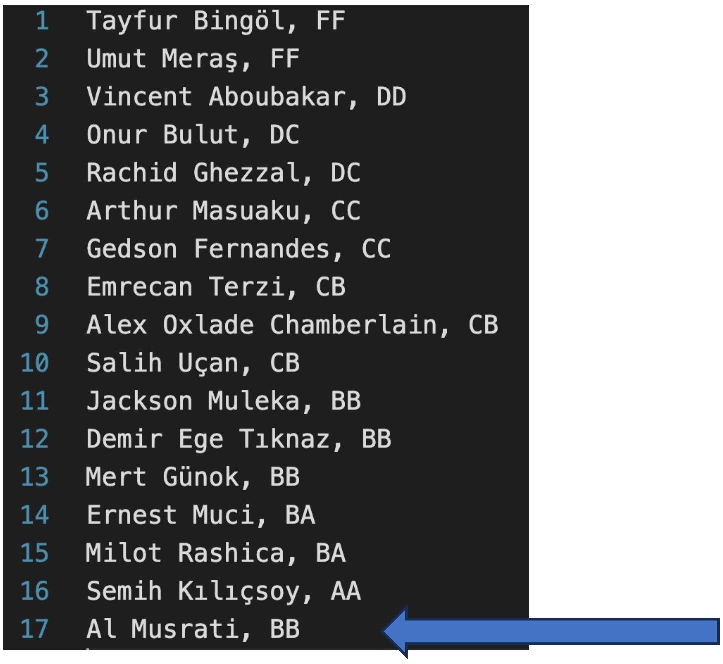


```
● (base) enesaysu@Enes-Aysu-MacBook-Air hw1 % ./gtuStudentGrades
  Type exit to terminate program.
  Enter command: gtuStudentGrades "grades.txt"
  "grades.txt" created successfully.
```

addStudentGrade “name surname” “grade” “grades.txt”

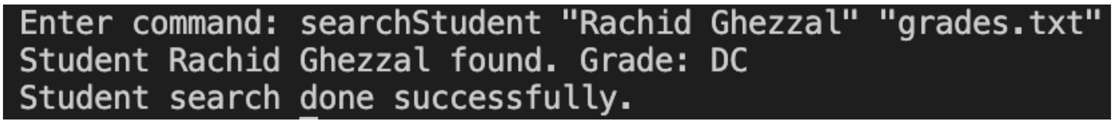


```
Enter command: addStudentGrade "Al Musrati" "BB" "grades.txt"
Student grade added successfully.
```



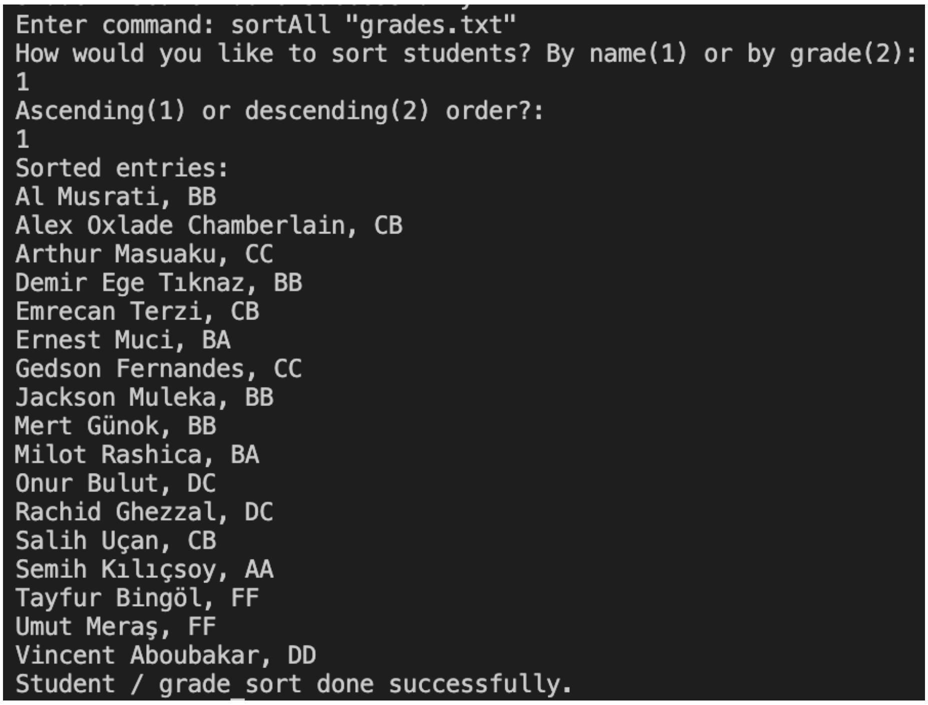
```
Tayfur Bingöl, FF
Umut Meraş, FF
Vincent Aboubakar, DD
Onur Bulut, DC
Rachid Ghezzal, DC
Arthur Masuaku, CC
Gedson Fernandes, CC
Emrecan Terzi, CB
Alex Oxlade Chamberlain, CB
Salih Uçan, CB
Jackson Muleka, BB
Demir Ege Tıknaz, BB
Mert Günok, BB
Ernest Muci, BA
Milot Rashica, BA
Semih Kılıçsoy, AA
Al Musrati, BB
```

searchStudent “name surname” “grades.txt”

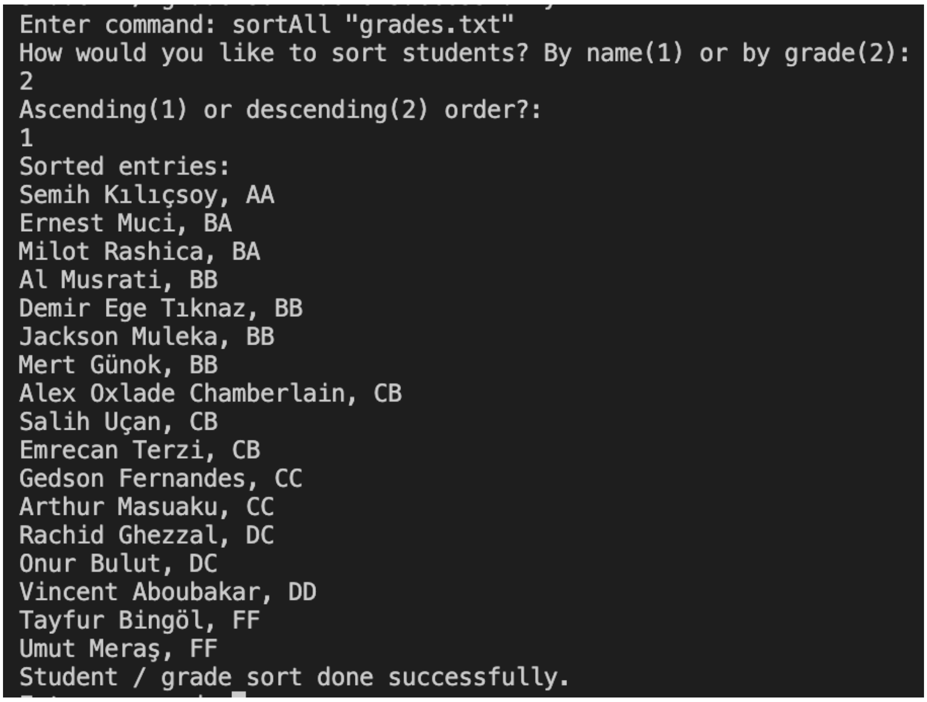


```
Enter command: searchStudent "Rachid Ghezzal" "grades.txt"
Student Rachid Ghezzal found. Grade: DC
Student search done successfully.
```

sortAll “grades.txt”

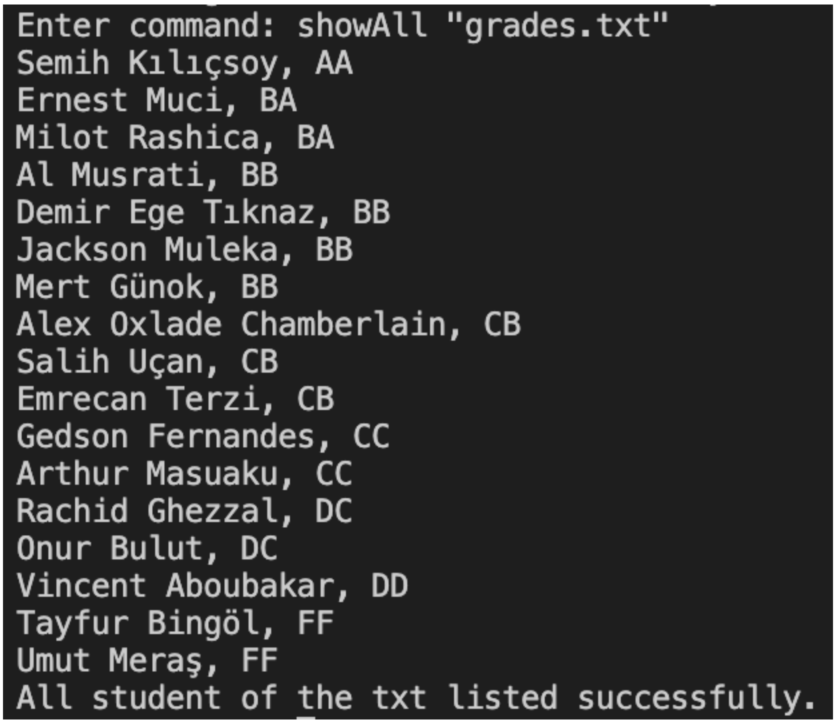


```
Enter command: sortAll "grades.txt"
How would you like to sort students? By name(1) or by grade(2):
1
Ascending(1) or descending(2) order?:
1
Sorted entries:
Al Musrati, BB
Alex Oxlade Chamberlain, CB
Arthur Masuaku, CC
Demir Ege Tıknaz, BB
Emrecan Terzi, CB
Ernest Muci, BA
Gedson Fernandes, CC
Jackson Muleka, BB
Mert Günok, BB
Milot Rashica, BA
Onur Bulut, DC
Rachid Ghezzal, DC
Salih Uçan, CB
Semih Kılıçsoy, AA
Tayfur Bingöl, FF
Umut Meraş, FF
Vincent Aboubakar, DD
Student / grade sort done successfully.
```



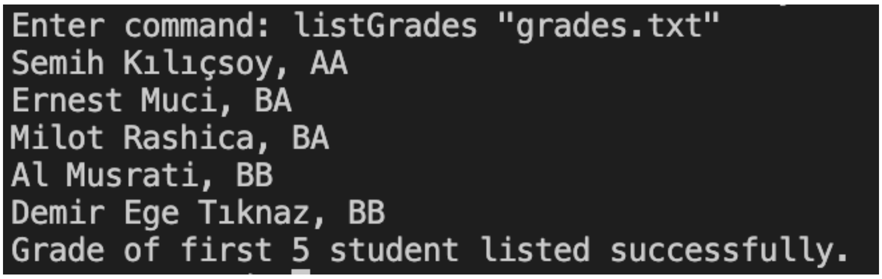
```
Enter command: sortAll "grades.txt"
How would you like to sort students? By name(1) or by grade(2):
2
Ascending(1) or descending(2) order?:
1
Sorted entries:
Semih Kılıçsoy, AA
Ernest Muci, BA
Milot Rashica, BA
Al Musrati, BB
Demir Ege Tıknaz, BB
Jackson Muleka, BB
Mert Günok, BB
Alex Oxlade Chamberlain, CB
Salih Uçan, CB
Emrecan Terzi, CB
Gedson Fernandes, CC
Arthur Masuaku, CC
Rachid Ghezzal, DC
Onur Bulut, DC
Vincent Aboubakar, DD
Tayfur Bingöl, FF
Umut Meraş, FF
Student / grade sort done successfully.
```

showAll “grades.txt”



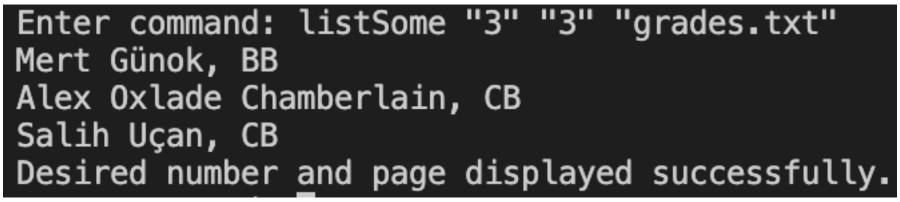
```
Enter command: showAll "grades.txt"
Semih Kılıçsoy, AA
Ernest Muci, BA
Milot Rashica, BA
Al Musrati, BB
Demir Ege Tıknaz, BB
Jackson Muleka, BB
Mert Günok, BB
Alex Oxlade Chamberlain, CB
Salih Uçan, CB
Emrecan Terzi, CB
Gedson Fernandes, CC
Arthur Masuaku, CC
Rachid Ghezzal, DC
Onur Bulut, DC
Vincent Aboubakar, DD
Tayfur Bingöl, FF
Umut Meraş, FF
All student of the txt listed successfully.
```

listGrades “grades.txt”



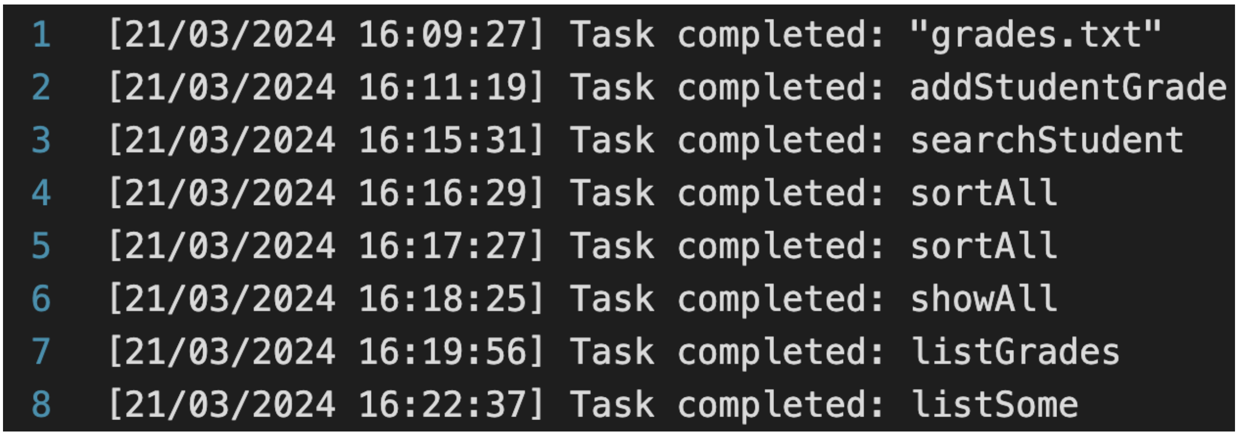
```
Enter command: listGrades "grades.txt"
Semih Kılıçsoy, AA
Ernest Muci, BA
Milot Rashica, BA
Al Musrati, BB
Demir Ege Tıknaz, BB
Grade of first 5 student listed successfully.
```

listSome “numOfEntries” “pageNumber” “grades.txt”



```
Enter command: listSome "3" "3" "grades.txt"
Mert Günok, BB
Alex Oxlade Chamberlain, CB
Salih Uçan, CB
Desired number and page displayed successfully.
```

Log File



```
[21/03/2024 16:09:27] Task completed: "grades.txt"
[21/03/2024 16:11:19] Task completed: addStudentGrade
[21/03/2024 16:15:31] Task completed: searchStudent
[21/03/2024 16:16:29] Task completed: sortAll
[21/03/2024 16:17:27] Task completed: sortAll
[21/03/2024 16:18:25] Task completed: showAll
[21/03/2024 16:19:56] Task completed: listGrades
[21/03/2024 16:22:37] Task completed: listSome
```

Summary

The provided program is a command-line utility for managing student grades. It offers various functionalities such as creating a grades file, adding student grades, searching for grades, sorting grades, and displaying grades. Each operation is handled asynchronously using child processes, enhancing the program's responsiveness. Additionally, the program logs all executed commands into a log file for reference. Overall, it provides a comprehensive solution for managing student grades efficiently.