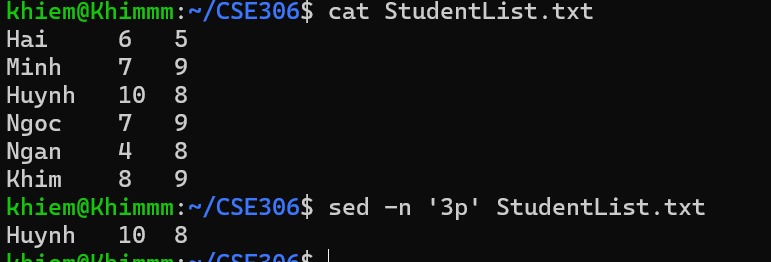
**Lab9\_NgoQuangKhiem\_2031220025**

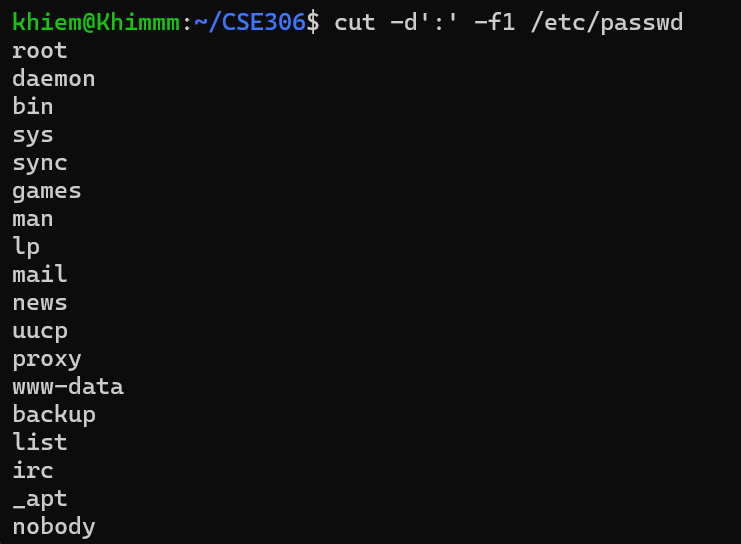
**1. Extract lines between two line numbers: example: 2 & 4 => extract line3.**

khiem@Khimmm:~/CSE306$ sed -n '3p' filename



**2. Write a command to print the first field in the passwd file**

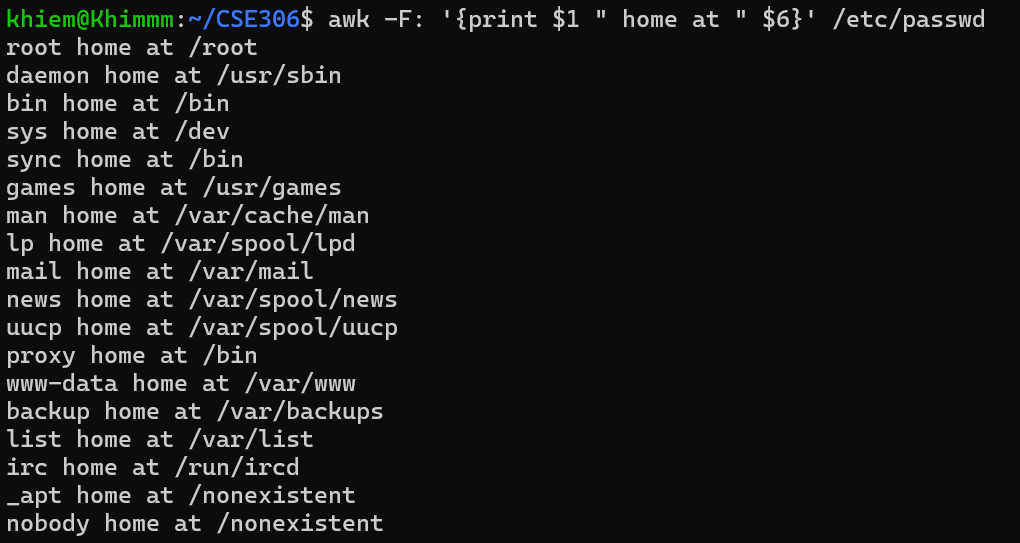
khiem@Khimmm:~/CSE306$ cut -d':' -f1 /etc/passwd

****

**3. Print the username and its home path from /etc/passwd**

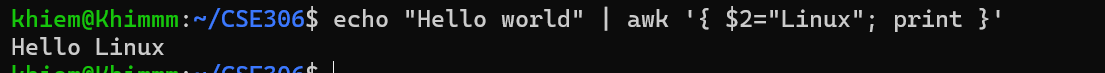
**ex: root home at /root**

awk -F: '{print $1 " home at " $6}' /etc/passwd

****

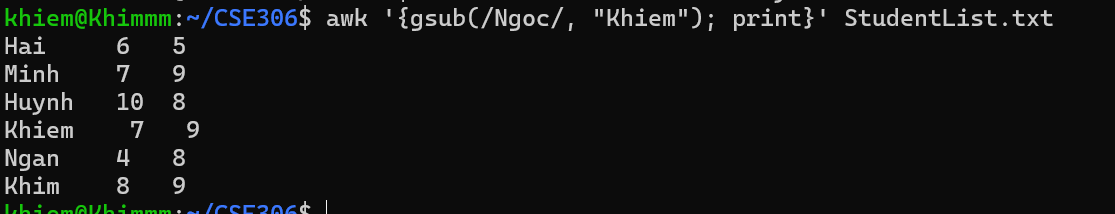
**4. Change the second word of a string ("Hello world") (=>using with echo command)**

khiem@Khimmm:~/CSE306$ echo "Hello world" | awk '{ $2="Linux"; print }'

****

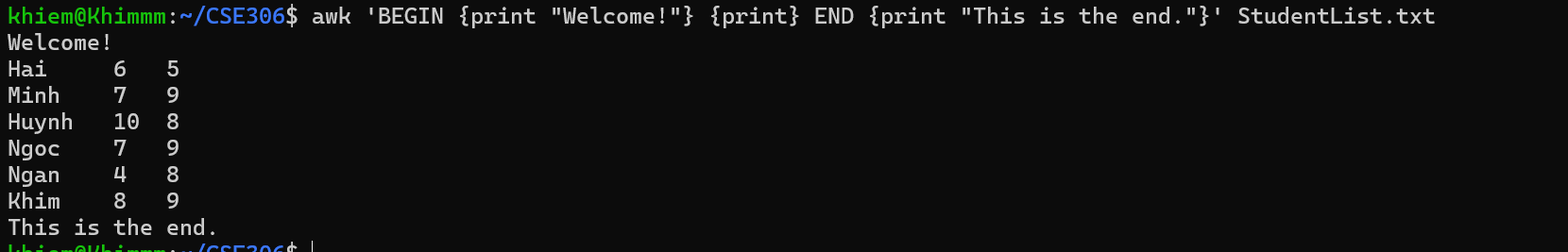
**5. Replace the name “Ngoc” into your name using gsub() command**

khiem@Khimmm:~/CSE306$ awk '{gsub(/Ngoc/, "Khiem"); print}' StudentList.txt

****

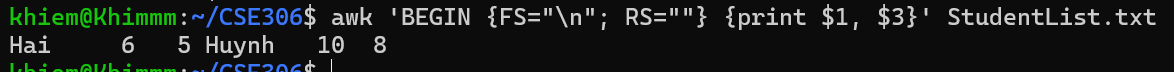
**6. Add the Header and the Footer for the content**

awk 'BEGIN {print "Welcome!"} {print} END {print "This is the end."}' StudentList.txt

****

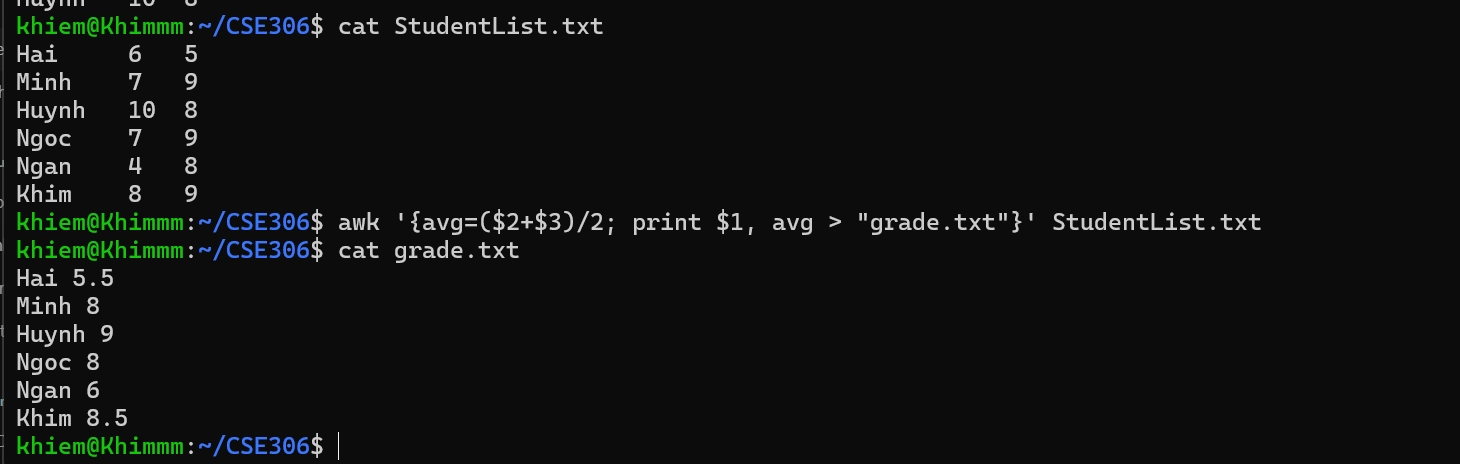
**7. Convert data from line format to column format, and print line 1 and line 3**

awk 'BEGIN {FS="\n"; RS=""} {print $1, $3}' StudentList.txt

****

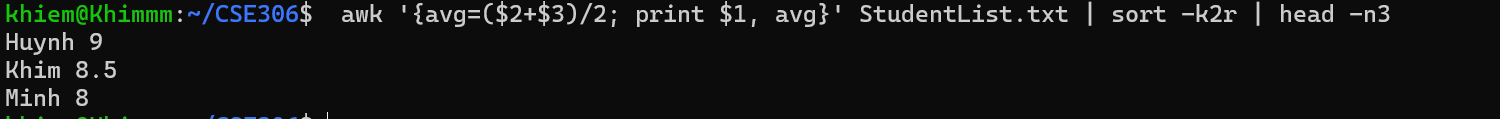
**8. Write a script to calculate average score and save into new file grade.txt**

awk '{avg=($2+$3)/2; print $1, avg > "grade.txt"}' StudentList.txt

****

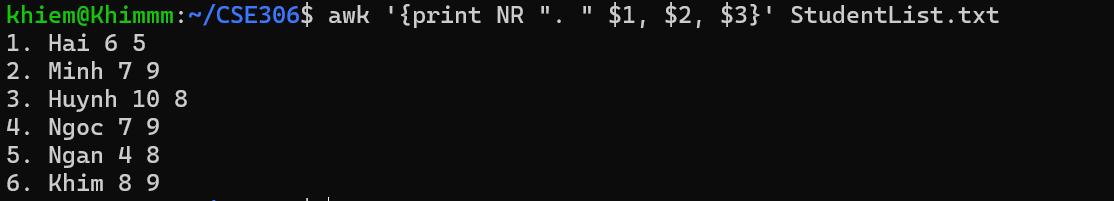
**9. Sort the average score list in descending order and show top 3**

awk '{avg=($2+$3)/2; print $1, avg }' StudentList.txt | sort -k2r | head -n3

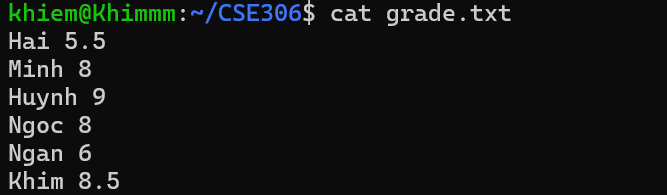
****

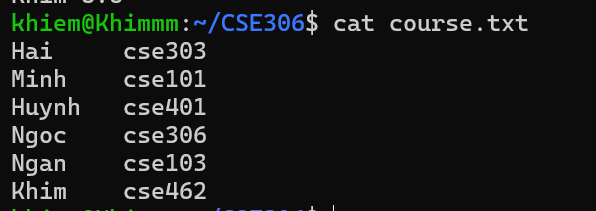
**10. Add ascending numbers before the name column**

awk '{print NR ". " $1, $2, $3}' StudentList.txt

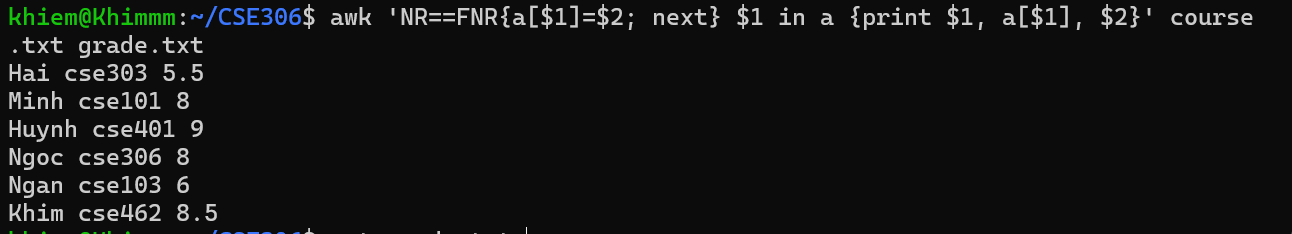
****

**11. Suppose there are 2 files: course.txt (student’s name, course ) and grade.txt (student’s name,grade), write a command to concatenate student names and their grades based on the course.**

****

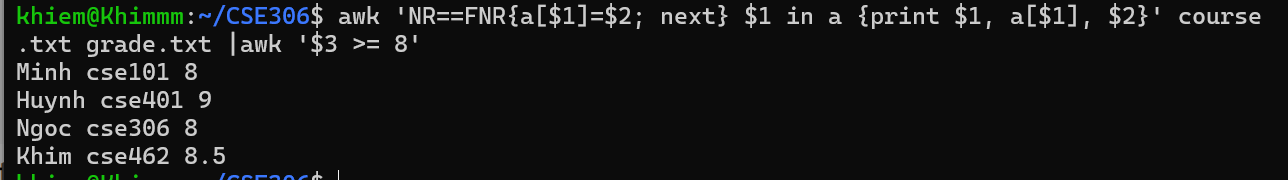
****

awk 'NR==FNR{a[$1]=$2; next} $1 in a {print $1, a[$1], $2}' course.txt grade.txt

****

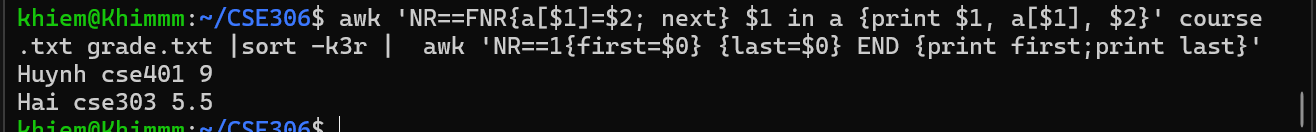
**12. Filter only students with the score >=8**

awk 'NR==FNR{a[$1]=$2; next} $1 in a {print $1, a[$1], $2}' course.txt grade.txt |awk '$3 >= 8'

****

**12. Print the student with the highest score and lowest ones.**

awk 'NR==FNR{a[$1]=$2; next} $1 in a {print $1, a[$1], $2}' course.txt grade.txt |sort -k3r | awk 'NR==1{first=$0} {last=$0} END {print first;print last}'

****

**13. Classify students by score (excellent, good, average, poor) => using if-else**

nano classify\_students.sh

—-----------------------------------------------------------------------------------------------

#!/bin/bash

# Check if the required files exist

if [[ ! -f course.txt || ! -f grade.txt ]]; then

echo "Error: course.txt or grade.txt not found!"

exit 1

fi

# Process files with awk to classify students by score

awk 'NR==FNR{a[$1]=$2; next} {score=$2; course=a[$1];

if(score >= 8) {grade="Excellent"}

else if(score >= 6.5) {grade="Good"}

else if(score >= 5) {grade="Average"}

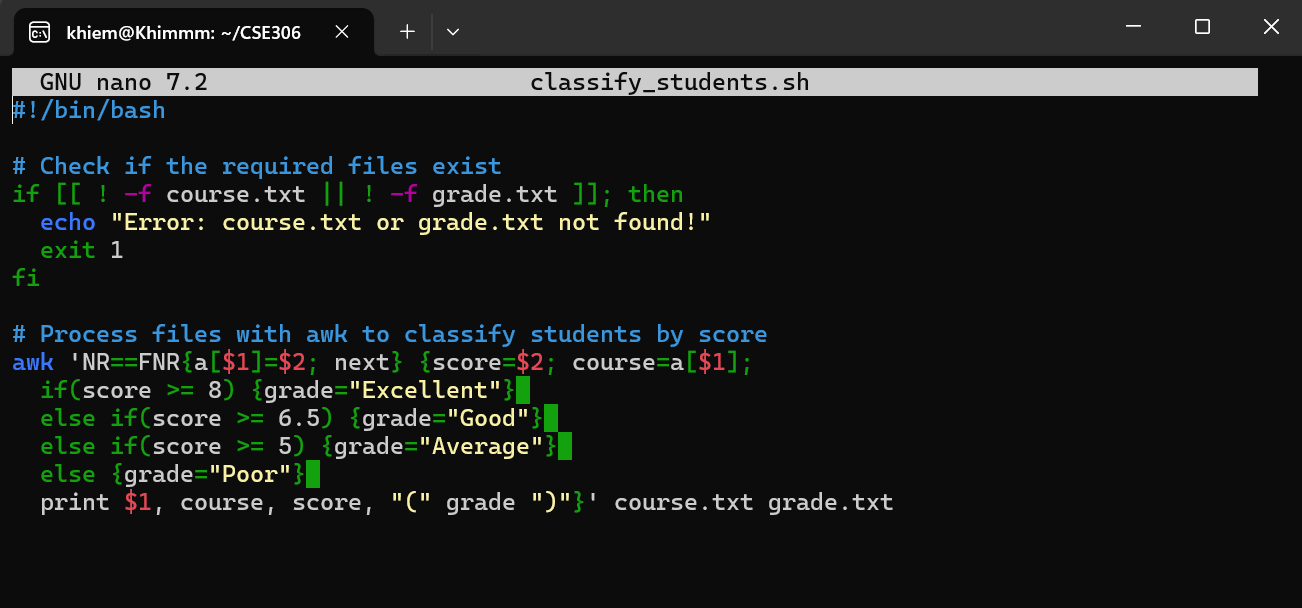
else {grade="Poor"}

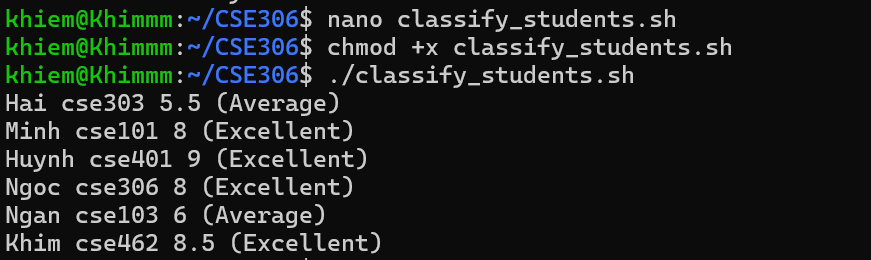
print $1, course, score, "(" grade ")"}' course.txt grade.txt

—---------------------------------------------------------------------------------------------------------

chmod +x classify\_students.sh

./classify\_students.sh

****

****

**14. Count the number of each category.**

nano Count\_each\_category.sh

—------------------------------------------------

#!/bin/bash

# Check if the required files exist

if [[ ! -f course.txt || ! -f grade.txt ]]; then

echo "Error: course.txt or grade.txt not found!"

exit 1

fi

# Initialize counters for each category

excellent\_count=0

good\_count=0

average\_count=0

poor\_count=0

# Process files with awk to classify students by score and count categories

awk 'NR==FNR{a[$1]=$2; next} {

score=$2;

course=a[$1];

if(score >= 8) {excellent\_count++}

else if(score >= 6.5) {good\_count++}

else if(score >= 5) {average\_count++}

else {poor\_count++}

}

END {

print "Excellent: " (excellent\_count ? excellent\_count : 0);

print "Good: " (good\_count ? good\_count : 0);

print "Average: " (average\_count ? average\_count : 0);

print "Poor: " (poor\_count ? poor\_count : 0)

}' course.txt grade.txt

—------------------------------------------------------------------------

chmod +x Count\_each\_category.sh

./Count\_each\_category.sh

