READ ME FILE FOR UNIQUE

Uniq.c description:

This code is a C program that implements a command for processing and filtering text data from either a file or standard input. The program supports various flags to perform different operations on the text data. Let's break down the code step by step:

The program includes necessary header files, such as "types.h", "stat.h", "user.h", and "fcntl.h", which likely contain definitions for system calls, data types, and file operations.

The emptyString function is a utility function that clears a character array by filling it with null characters ('\0').

The program defines several functions, each of which handles a specific operation based on different command-line flags:

pipeFlag(char *argv[]): Handles the case when no flags are provided as command-line arguments. It reads and processes lines from standard input and prints unique lines to the standard output. It uses buffer1, buffer2, and temp as temporary storage for line comparison.

noFlag(char *argv[]): Handles the case when a filename is provided without any flags. It reads and processes lines from the specified file and prints unique lines to the standard output.

countFlag(char *argv[]): Handles the case when the -c flag is provided. It reads and processes lines from the specified file and prints unique lines along with their counts.

duplicateFlag(char *argv[]): Handles the case when the -d flag is provided. It reads and processes lines from the specified file and prints duplicate lines.

ignoreFlag(char *argv[]): Handles the case when the -i flag is provided. It reads and processes lines from the specified file, converts all characters to lowercase, and prints unique lines to the standard output.

toLowerCase(char *str): A utility function to convert a string to lowercase.

In the main function, the program checks the number of command-line arguments (argc) to determine which case to execute:

If argc is 1, it calls pipeFlag, which reads and processes lines from standard input.

If argc is less than or equal to 2, it calls noFlag, which reads and processes lines from a specified file.

If argc is 3 and a valid flag is provided (e.g., -c, -d, -i), it calls the corresponding flag-specific function (e.g., countFlag, duplicateFlag, ignoreFlag) based on the provided flag.

If an invalid flag is provided, it prints an error message.

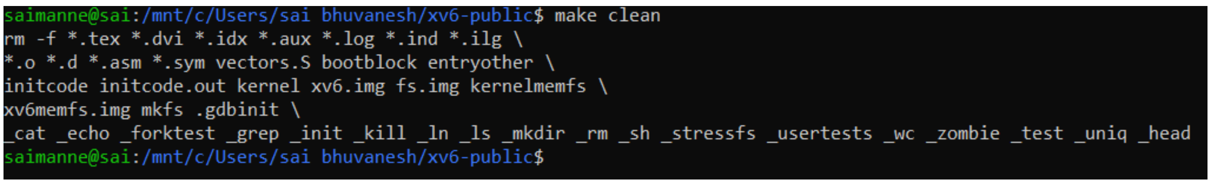
The program performs text processing and filtering based on the chosen operation and flag, and the results are printed to the standard output.

Steps for execution:

1. 1.we use wsl command to go to ubuntu.

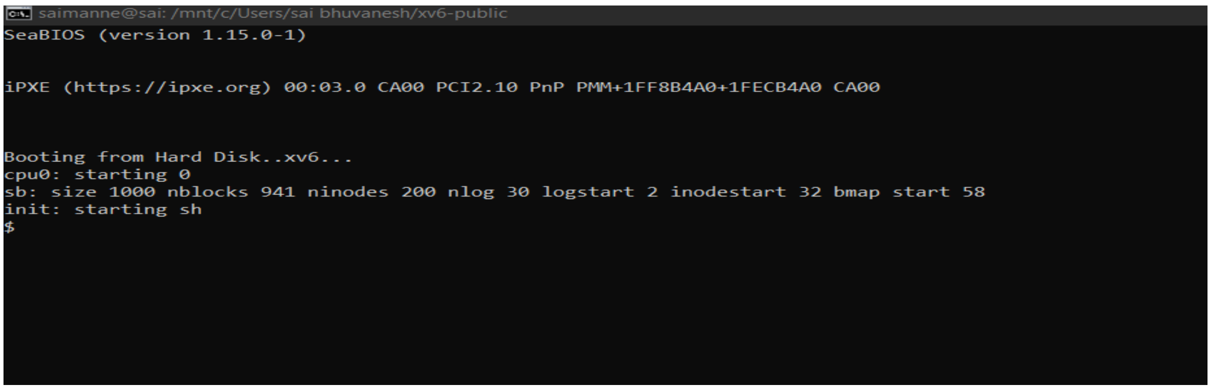
2.we change the directory to cd xv6-public

3.we use the command make clean



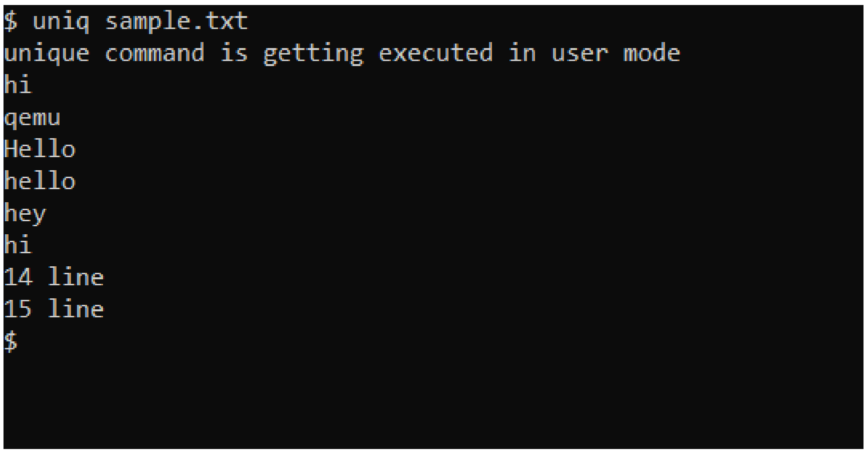
```
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$ make clean
rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
*.o *.d *.asm *.sym vectors.S bootblock entryother \
initcode initcode.out kernel xv6.img fs.img kernelmemfs \
xv6memfs.img mkfs .gdbinit \
_cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs _usertests _wc _zombie _test _uniq _head
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$
```

4.we use the command make qemu-nox



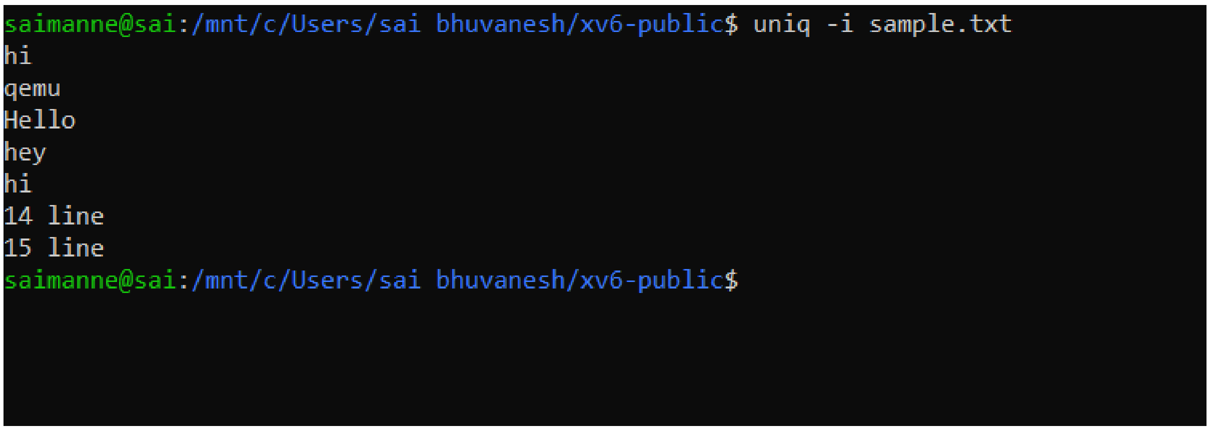
```
saimanne@sai: /mnt/c/Users/sai bhuvanesh/xv6-public
SeaBIOS (version 1.15.0-1)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B4A0+1FECB4A0 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$
```

5. to print the uniq lines in the files we use the command uniq “file name”



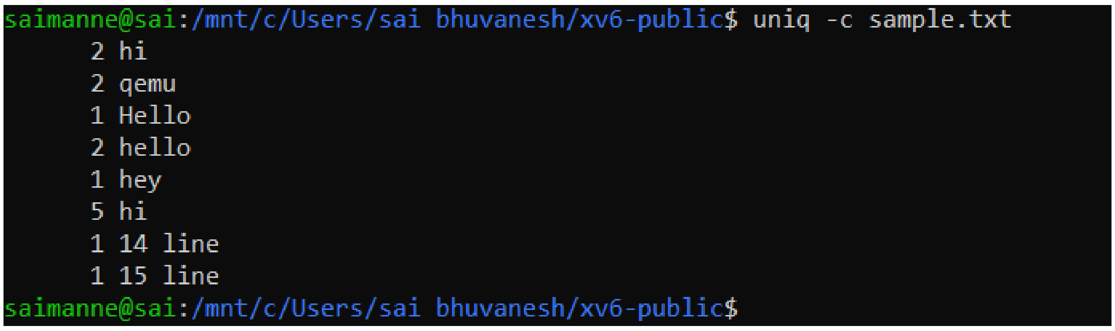
```
$ uniq sample.txt
unique command is getting executed in user mode
hi
qemu
Hello
hello
hey
hi
14 line
15 line
$
```

6.to ignore cases we use command uniq -i “file name”



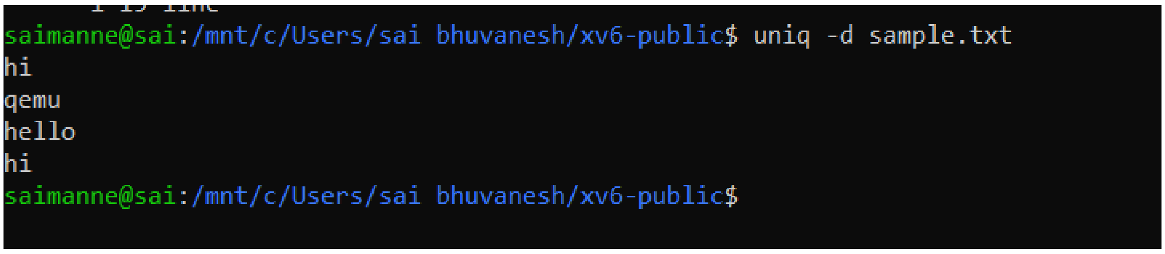
```
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$ uniq -i sample.txt
hi
qemu
Hello
hey
hi
14 line
15 line
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$
```

7.to count the number of times a line is repeated in the given file we use the command uniq -c “file name”.



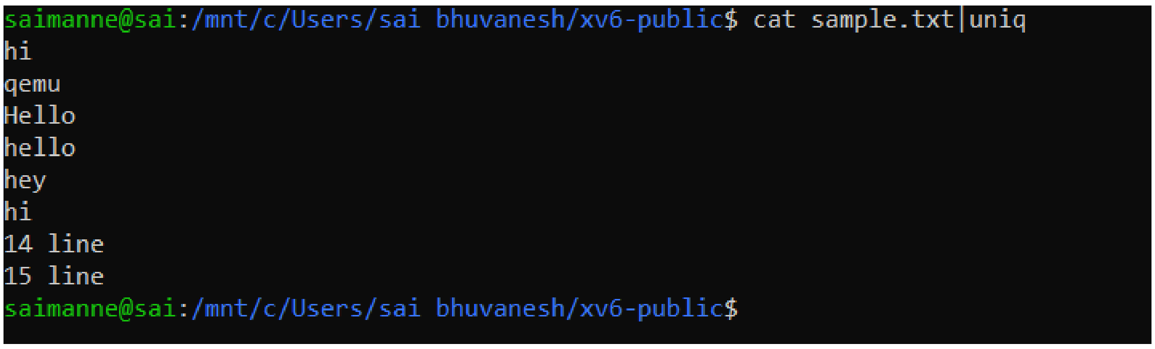
```
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$ uniq -c sample.txt
      2 hi
      2 qemu
      1 Hello
      2 hello
      1 hey
      5 hi
      1 14 line
      1 15 line
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$
```

8.to print only the duplicate lines in the files we use the command uniq -d “file name”



```
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$ uniq -d sample.txt
hi
qemu
hello
hi
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$
```

9.to print uniq lines from the file using pipes we use the command cat “filename”|uniq



```
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$ cat sample.txt|uniq
hi
qemu
Hello
hello
hey
hi
14 line
15 line
saimanne@sai:/mnt/c/Users/sai bhuvanesh/xv6-public$
```

.

i have tried implementing the above two codes in kernal mode but im getting errors .that is why I have not implemented it in kernal mode.