**UNIQ – KERNEL LEVEL**

**Uniq.c:**

* For kernel mode execution enable uniq() function call in uniq.c and disable printText() call in uniq.c. The main function is same.
* In main comment out the lines which makes use of cmnd. Instead of cmnd I am using a char and passing its address to the uniq() function call.
* In uniq.c main function :

1. Commnet out lines with make use of cmnd

(i.e 110, 118,135 line numbers).

2. enable lines 111,119,136).

A screen shot of a computer program

Description automatically generated

**For kernel level Implementation:**

I have made changes in the following files:

1. Syscall.c

A screen shot of a computer

Description automatically generated

A black background with white text

Description automatically generated

1. user.h



1. defs.h



1. usys.S

A black rectangle with white text

Description automatically generated

1. syscall.h

A screenshot of a computer

Description automatically generated

1. sysfile.c

* In sysfile.c I have implemented the function sys\_uniq which retrives the parameters that are passed for uniq() in main function in uniq.c.
* I have made use of struct file \* and char \* and retrieved the input parameters and then called the function uniq and passed these parameters.
* I have used functions argfd() to retrieve the file pointer and argstr() to retrieve the string parameter.

A computer screen shot of white text

Description automatically generated

1. proc.c

* Within proc.c I have implemented the same logic similar to uniq.c but instead of dynamic memory allocation to struct. I have defined it statically.
* I have taken to char arrays current\_line and prev\_line and read each character of the file and stored it in current\_line.
* But here, instead of strcmp I have used strncmp which takes three parameters prev\_line, current\_line, and the bufsize.
* The printLines() and the strcasecmp() functions are same as in uniq.c.
* Once that printlines() is called and implementes after that we are using memset to free the prev\_line.
* Once the function is completed we are using memset to free the current\_line and prev\_line and setting them to 0.

**EXECUTION:**

**Textfile: OS.txt**

**Makefile:**

1. For executing, I have made changes to the makefile. I have placed uniq.c under EXTRA,

\_uniq\ under UPROGS and within the below lines in Makefile.c

fs.img: mkfs README uniq.c $(UPROGS)

./mkfs fs.img README OS.txt $(UPROGS)

2. Then in the Terminal go to the xv6-public directory and then execute

make clean

make

make qemu

3. This will open the qemu.

4. With in the qemu execute:

**cat OS.txt|uniq**

This will print the uniq lines of the text file

**uniq -c OS.txt**

This will print the first no of times the line is duplicated along with the duplicated line.

**uniq -d OS.txt**

this will print only the line that are repeated

**uniq -i OS.txt**

this will print the uniq lines without considering the case sensitivity.

The below is the screenshot of execution.

A screenshot of a computer

Description automatically generated

**Resources Used:**

**Group Members:**

1. Gnana Sreeram Jogi
2. Akshitha Maddipatla

TA:

1.Manas

**TA:**

1.Manas

**Online Resources:** for kernel level system calls.

https://www.cse.iitd.ernet.in/~sbansal/os/previous\_years/2011/xv6\_html/syscall\_8c-source.html#l00124