Implementation of uniq command

Implemented in XV6-RISCV in DOCKER (in Mac OS - M1)

Code logic:

File: **uniq.c**

All the variables are declared. If 1 or less argument is given, the standard input is taken and passed to the uniq_f function and the uniq system call. var_icd contains -I/-c/-d if given, else it is defaulted to a random character o. Appropriate error message is printed on file read error.

/xv6-riscv/user/uniq.c C

```c
int main(int argc, char *argv[]) {
    char buffer[512];
    int n;
    int fdk = 0, fd=0;
    char a;
    char var_icd = 'o';
    int f=0;

    if (argc <= 1) {
        uniq_f(0,var_icd);

        while((n = read(0, buffer, sizeof(buffer))) > 0 )
        {
           uniq(0, buffer, var_icd);
        }
        if (n < 0) {
            printf("Error: Unable to read file(Kernel mode)\n");
        }
        exit(0);
```

The command arguments are read, if '-i/c/d' is provided it is stored in *var_icd* variable. A flag *f* is set to 1 if -i/c/d is provided.

/xv6-riscv/user/uniq.c C

```c
    } else {
          for (int i = 1; i < argc; i++) {
            a=*argv[i];
            if(a!='-'){
                fd = open(argv[i], 0);
                if (fd < 0) {
                  printf("uniq: cannot open %s\n", argv[i]);
                  exit(1);
                }
            }

          if(a=='-'){
              argv[i]++;
              var_icd = *argv[i];
              f=1;
          }
```

If -c/i/d is not given, the arguments with default *var_icd* value is passed to the function and system call. For system call *uniq,* the file is read and the content stored in the variable *buffer* is passed.

/xv6-riscv/user/uniq.c C

```c
            // when -c/i/d is not given
            if(f==0){
                uniq_f(fd,var_icd);
                fdk = open(argv[i], 0);
                while((n = read(fdk, buffer, sizeof(buffer))) > 0 )
                {
                    uniq(n, buffer, var_icd);
                }
                if (n < 0) {
                    printf("Error: Unable to read file(kernel mode)\n");
                }
                    close(fdk);
                    close(fd);
            }
            f=0;
        }
    }
    exit(0);
}
```

uniq_f function:

The file is read, and the current line is stored in *temp_curr* variable

/xv6-riscv/user/uniq.c C

```c
#include "kernel/types.h"
#include "kernel/stat.h"
#include "user/user.h"

char buf[1024];

int uniq_f(int fd, char var_icd) {
    int n, i, j=0, f=0 , cnt = 1;
    char var_ic='a', var_ip='a';
    char temp_curr[1024] = "";
    char temp_prev[1024] = "";

    n = read(fd, buf, sizeof(buf));
    printf("Uniq command is getting executed in user mode.\n");

    for (i = 0; i < n; i++) {
        if (buf[i] != '\n') {
            temp_curr[j] = buf[i];
            j += 1;
```

When ‘-i’ is passed as an argument which is stored in var_icd variable, the previous and the current line’s each character is converted into lowercase and compared. A flag ‘f’ is set to 1, and break; is used to end the comparison if there is any mismatch.

/xv6-riscv/user/uniq.c

```
22              temp_curr[j]='\n';
23              if(var_icd != 'c' && var_icd != 'd'){
24                   if (strcmp(temp_curr, temp_prev) != 0){
25                       if(var_icd == 'i'){
26                           int k = 0;
27                           //to convert line to lowercase to compare when -i is given
28                           while(temp_curr[k] != '\n' && temp_prev[k] != '\n'){
29                               if(temp_curr[k] >= 'A' && temp_curr[k] <= 'Z')
30                                   var_ic = temp_curr[k] + 32;
31                               else
32                                   var_ic = temp_curr[k];
33                               if(temp_prev[k] >= 'A' && temp_prev[k] <= 'Z')
34                                   var_ip = temp_prev[k] + 32;
35                               else
36                                   var_ip = temp_prev[k];
37                               if(var_ic == var_ip){
38                                   k+=1;
39                               }
40                              else{
41                                f = 1;
42                                break;
43                              }
```

If lines are equal, current line is printed and copied to the previous line variable, to be available to store the next line.

/xv6-riscv/user/uniq.c C

```
43                              }
44                          }
45                          if(f==1){ //lines are not equal
46                              strcpy(temp_prev, temp_curr);
47                              printf(temp_curr);
48                          }
49                          f=0;
50                      }else{ //when -i is not given
51                          strcpy(temp_prev, temp_curr);
52                          printf(temp_curr);
53                      }
54
55                  }
56              }
```

If -c is passed as argument, the total count of repeating lines should be printed with the lines. if the first line of file is read I,e. Previous line is empty, then current line is copied to the previous line variable to be available to store the next line. If *temp_prev* is not empty, then, the count *cnt* is

incremented. When the lines are unequal, the previous line is printed with the count and the count is reset to 1. I > n-2 I,e is used to determine the last line and print.

/xv6-riscv/user/uniq.c C

```c
                }
                if(var_icd == 'c'){
                    if(temp_prev[0] != '\0'){
                        if(strcmp(temp_curr, temp_prev) == 0)
                            cnt++;
                        else{
                            printf("%d %s", cnt, temp_prev);
                            cnt = 1;
                        }
                    }
                    strcpy(temp_prev, temp_curr);
                    if(i > n-2){ //check last line
                        printf("%d %s", cnt, temp_curr);
                    }
```

If -d is given: Only the repeating lines should be printed. If the lines are equal, the count is incremented. Once the lines are unequal, the count is checked, only if it is greater than 1, the line is printed. The current line is copied to the previous line variable to be available to store the next line. The current line variable *temp_curr* and other variables are cleared. An error is printed if there is any file read error.

/xv6-riscv/user/uniq.c

```c
            }else if(var_icd == 'd'){
                if(strcmp(temp_curr, temp_prev) == 0)
                    cnt++;
                else{
                    if(cnt > 1)
                    printf("%s", temp_prev);
                    strcpy(temp_prev, temp_curr);
                    cnt = 1;
                }
                if(i > n-2 && cnt > 1){ //check last line
                    printf("%s",temp_curr);
                }
            }

            memset(temp_curr, 0, sizeof(temp_curr));
            j=0;
        }
    }
    if (n < 0) {
        printf("uniq: read error\n");
        exit(1);
    }
     return 0;
```

File: sysproc.c (kernel mode implementation)

Data declarations and arguments are retrieved into the respective variables.

/xv6-riscv/kernel/sysproc.c

```c
//Custom uniq system call
uint64
sys_uniq(void)
{
  int n,i;
  int j = 0, k = 0, f = 0, l = 0;
  int eq = 0, cnt = 1;
  char buffer[512];
  int var_icd;
  char temp_curr[512] = "";
  char temp_prev[512] = "";
  char var_ic = 'a';
  char var_ip = 'a';

  argint(0, &n);
  argstr(1, buffer, sizeof(buffer));
  argint(2, &var_icd);  //-c or -d or -i ; by default dummy char o is passed
```

For loop is used to iterate over the file content that is passed as an argument to the system call. Each line is stored in a *temp_curr* variable.

```c
  printf("\nUniq command is getting executed in kernel mode\n");
  for(i=0;i<=n ;i++)
    {
      if(buffer[i]!='\n'){
          temp_curr[j] = buffer[i];
          j+=1;
      }
      else{
          temp_curr[j]='\n';
```

temp_prev is stored with the previous line. The *temp_prev* and *temp_curr* I,e. The current and the previous line is compared.

When '*-i*' is passed as an argument which is stored in *var_icd variable,* the previous and the current line's each character is converted into lowercase and compared. A flag '*f*' is set to 1, and *break;* is used to end the comparison if there is any mismatch.

/xv6-riscv/kernel/sysproc.c

```
121             //compare previous and current statement
122             while(temp_curr[k] != '\n' || temp_prev[k] != '\n'){
123                 if(temp_curr[k] == temp_prev[k])
124                     k+=1;
125                 else if(var_icd == 'i' && temp_curr[k] != '\n' && temp_prev[k] != '\n'){
126                     //convert string to lowercase to compare when -i
127                     if(temp_curr[k] >= 'A' && temp_curr[k] <= 'Z'){
128                         var_ic = temp_curr[k] + 32;
129                     }else
130                         var_ic = temp_curr[k];
131                     if(temp_prev[k] >= 'A' && temp_prev[k] <= 'Z'){
132                         var_ip = temp_prev[k] + 32;
133                     }else
134                         var_ip = temp_prev[k];
135                     if(var_ic == var_ip){
136                         k+=1;
137                     }
138                     else{
139                         f = 1;
140                         break;
141                     }
142                 }else{
143                   f = 1;
144                   break;
```

If flag *f* is 0 I,e. The lines are same, then a *eq* flag is set to 1, Indicating the previous and the current lines are same.

When '*-c'* or '*-d*' is not given as argument: If the lines are not equal (*eq* = 0), the current line *temp_curr* is copied to the *temp_prev* variable, to be available to store the next line. The current line is printed.

/xv6-riscv/kernel/sysproc.c

```
146                 }
147             if(f == 0) //equal lines
148                 eq = 1;
149             if(var_icd != 'c' && var_icd != 'd'){
150                 if(eq == 0){ //lines not equal
151                     for(l=0;temp_curr[l]!='\0';l++){
152                       temp_prev[l] = temp_curr[l];
153                     }
154                     temp_prev[l]='\0';
155                     printf("%s",temp_curr);
156                 }
```

If -c is given, The total count of repeating lines should be printed with the lines. if the first line of file is read I,e. Previous line is empty, then current line is copied to the previous line variable to be available to store the next line. If *temp_prev* is not empty, then, if the lines are equal (eq = 1), the count *cnt* is incremented. One the lines are unequal, the previous line is printed with the count.

/xv6-riscv/kernel/sysproc.c

```
156             }
157         }else if(var_icd == 'c'){
158             if(temp_prev[0] != '\0'){
159                 if(eq == 1)
160                   cnt++;
161                 else{
162                   printf("%d %s", cnt, temp_prev);
163                   cnt = 1;
164                 }
165               }
166               //copy current line to a variable
167               for(l=0;temp_curr[l]!='\0';l++){
168                   temp_prev[l] = temp_curr[l];
169               }
170               temp_prev[l]='\0';
171               if(i > n-2){ //print last line
172                   printf("%d %s", cnt, temp_curr);
173               }
```

If -d is given: Only the repeating lines should be printed. If the lines are equal (eq = 1), the count is incremented. Once the lines are unequal, the count is checked, if it is greater than 1, the line is

/xv6-riscv/kernel/sysproc.c

```
174         }else if(var_icd == 'd'){
175               if(eq == 1)
176                   cnt++;
177               else{
178                   if(cnt > 1)
179                       printf("%s", temp_prev);
180                   for(l=0;temp_curr[l]!='\0';l++){
181                       temp_prev[l] = temp_curr[l];
182                   }
183                   temp_prev[l]='\0';
184                   cnt = 1;
185               }
186               if(i > n-2 && cnt > 1){
187                   printf("%s",temp_curr);
188               }
189           }
190
```

printed. The current line is copied to the previous line variable to be available to store the next line.

/xv6-riscv/kernel/sysproc.c

```
190
191           //clear current line variable
192           memset(temp_curr, 0, sizeof(temp_curr));
193           j=0;
194           k=0;
195           eq=0;
196           f=0;
197         }
198       }
199        return 0;
200   }
```

The current line variable *temp_curr* and the other variables are cleared.

Other files modified:

/xv6-riscv/kernel/syscall.h

```
 6  #define SYS_read    5
 7  #define SYS_kill    6
 8  #define SYS_exec    7
 9  #define SYS_fstat   8
10  #define SYS_chdir   9
11  #define SYS_dup    10
12  #define SYS_getpid 11
13  #define SYS_sbrk   12
14  #define SYS_sleep  13
15  #define SYS_uptime 14
16  #define SYS_open   15
17  #define SYS_write  16
18  #define SYS_mknod  17
19  #define SYS_unlink 18
20  #define SYS_link   19
21  #define SYS_mkdir  20
22  #define SYS_close  21
23  #define SYS_uniq   22
24  #define SYS_head   23
```

Syscall.h

/xv6-riscv/user/usys.pl

```
23  entry("write");
24  entry("close");
25  entry("kill");
26  entry("exec");
27  entry("open");
28  entry("mknod");
29  entry("unlink");
30  entry("fstat");
31  entry("link");
32  entry("mkdir");
33  entry("chdir");
34  entry("dup");
35  entry("getpid");
36  entry("sbrk");
37  entry("sleep");
38  entry("uptime");
39  entry("uniq");
40  entry("head");
```

usys.pl

Sys call.c

/xv6-riscv/kernel/syscall.c

```
96   extern uint64 sys_uptime(void);
97   extern uint64 sys_open(void);
98   extern uint64 sys_write(void);
99   extern uint64 sys_mknod(void);
100  extern uint64 sys_unlink(void);
101  extern uint64 sys_link(void);
102  extern uint64 sys_mkdir(void);
103  extern uint64 sys_close(void);
104  extern uint64 sys_uniq(void);
105  extern uint64 sys_head(void);
106
```

/xv6-riscv/kernel/syscall.c

```
121  [SYS_sbrk]    sys_sbrk,
122  [SYS_sleep]   sys_sleep,
123  [SYS_uptime]  sys_uptime,
124  [SYS_open]    sys_open,
125  [SYS_write]   sys_write,
126  [SYS_mknod]   sys_mknod,
127  [SYS_unlink]  sys_unlink,
128  [SYS_link]    sys_link,
129  [SYS_mkdir]   sys_mkdir,
130  [SYS_close]   sys_close,
131  [SYS_uniq]    sys_uniq,
132  [SYS_head]    sys_head,
```

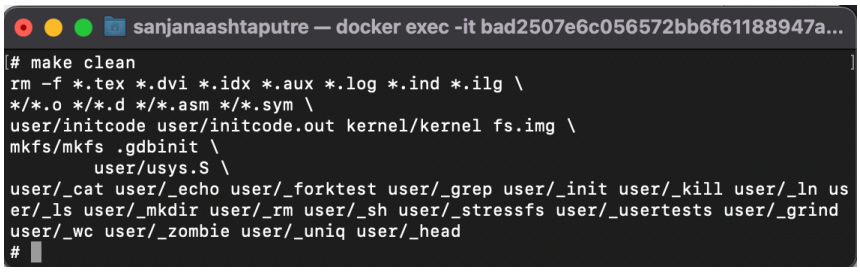
User.h

/xv6-riscv/user/user.h

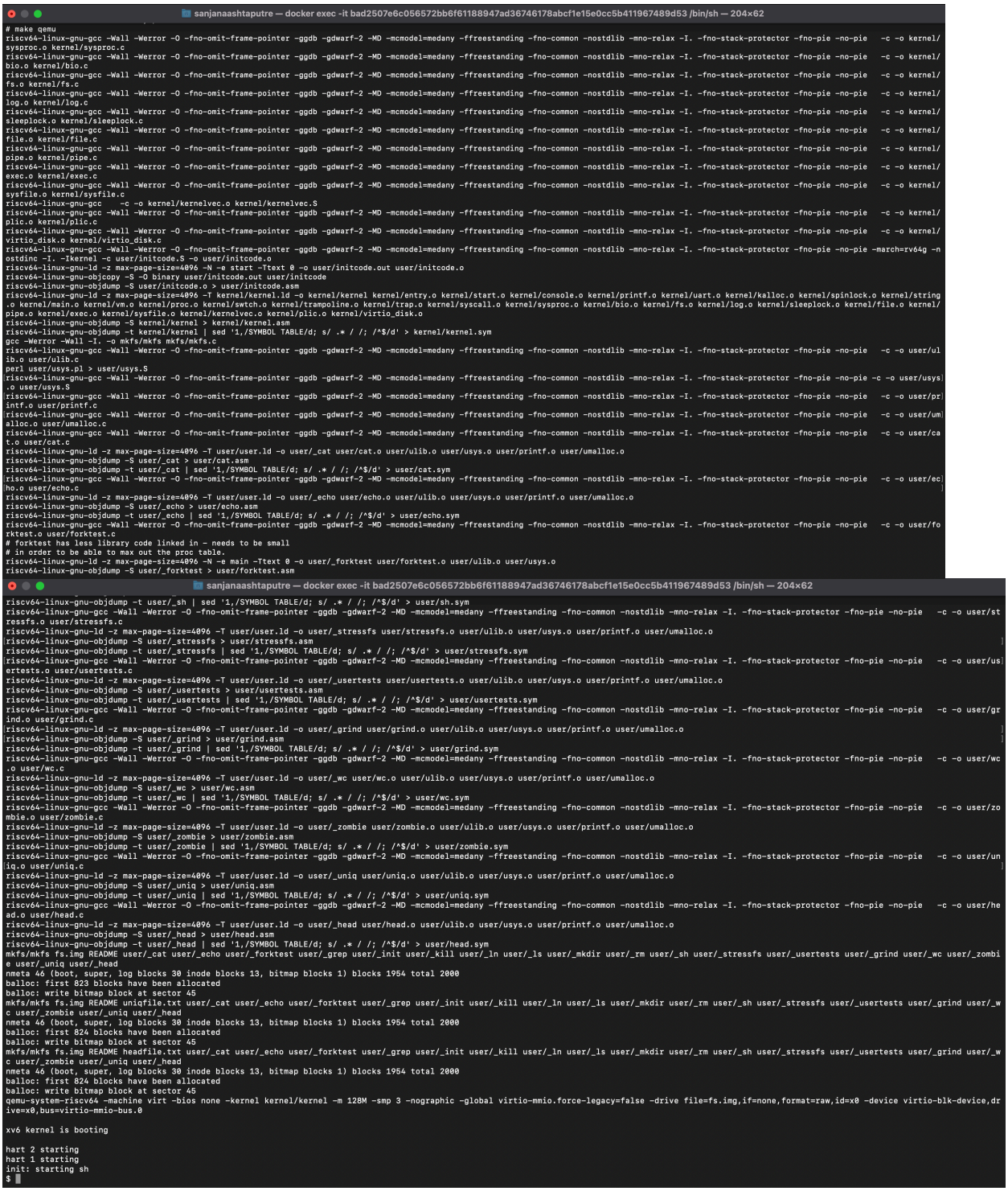
```
18  int mkdir(const char*);
19  int chdir(const char*);
20  int dup(int);
21  int getpid(void);
22  char* sbrk(int);
23  int sleep(int);
24  int uptime(void);
25  int uniq(int n, char *buffer, char var_icd);
26  int head(int n, char *buffer, int lines);
```

Output with screenshots of each step:

make clean

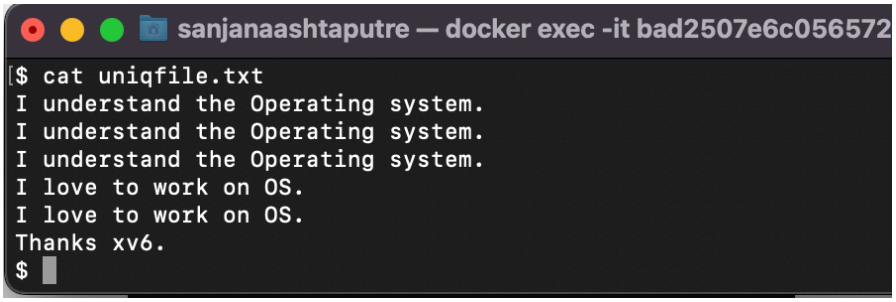


Make qemu


sanjanaashtaputre — docker exec -it bad2507e6c056572bb6f61188947ad36746178abcf1e15e0cc5b411967489d53 /bin/sh — 204×62
make qemu
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/sysproc.o kernel/sysproc.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/bio.o kernel/bio.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/fs.o kernel/fs.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/log.o kernel/log.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/sleeplock.o kernel/sleeplock.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/file.o kernel/file.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/pipe.o kernel/pipe.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/exec.o kernel/exec.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/sysfile.o kernel/sysfile.c
riscv64-linux-gnu-gcc -c -o kernel/kernelvec.o kernel/kernelvec.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/plic.o kernel/plic.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/virtio_disk.o kernel/virtio_disk.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -march=rv64g -nostdinc -I. -Ikernel -c user/initcode.S -o user/initcode.o
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e start -Ttext 0 -o user/initcode.out user/initcode.o
riscv64-linux-gnu-objcopy -S -O binary user/initcode.out user/initcode
riscv64-linux-gnu-objdump -S user/initcode.o > user/initcode.asm
riscv64-linux-gnu-ld -z max-page-size=4096 -T kernel/kernel.ld -o kernel/kernel kernel/entry.o kernel/start.o kernel/console.o kernel/printf.o kernel/uart.o kernel/kalloc.o kernel/spinlock.o kernel/string.o kernel/main.o kernel/vm.o kernel/proc.o kernel/swtch.o kernel/trampoline.o kernel/trap.o kernel/syscall.o kernel/sysproc.o kernel/bio.o kernel/fs.o kernel/log.o kernel/sleeplock.o kernel/file.o kernel/pipe.o kernel/exec.o kernel/sysfile.o kernel/kernelvec.o kernel/plic.o kernel/virtio_disk.o
riscv64-linux-gnu-objdump -S kernel/kernel > kernel/kernel.asm
riscv64-linux-gnu-objdump -t kernel/kernel | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > kernel/kernel.sym
gcc -Werror -Wall -I. -o mkfs/mkfs mkfs/mkfs.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/ulib.o user/ulib.c
perl user/usys.pl > user/usys.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/usys.o user/usys.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/printf.o user/printf.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/umalloc.o user/umalloc.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/cat.o user/cat.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_cat user/cat.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_cat > user/cat.asm
riscv64-linux-gnu-objdump -t user/_cat | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/cat.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/echo.o user/echo.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_echo user/echo.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_echo > user/echo.asm
riscv64-linux-gnu-objdump -t user/_echo | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/echo.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/forktest.o user/forktest.c
forktest has less library code linked in - needs to be small
in order to be able to max out the proc table.
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_forktest user/forktest.o user/ulib.o user/usys.o
riscv64-linux-gnu-objdump -S user/_forktest > user/forktest.asm
sanjanaashtaputre — docker exec -it bad2507e6c056572bb6f61188947ad36746178abcf1e15e0cc5b411967489d53 /bin/sh — 204×62
riscv64-linux-gnu-objdump -t user/_sh | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/sh.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/stressfs.o user/stressfs.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_stressfs user/stressfs.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_stressfs > user/stressfs.asm
riscv64-linux-gnu-objdump -t user/_stressfs | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/stressfs.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/usertests.o user/usertests.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_usertests user/usertests.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_usertests > user/usertests.asm
riscv64-linux-gnu-objdump -t user/_usertests | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/usertests.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/grind.o user/grind.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_grind user/grind.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_grind > user/grind.asm
riscv64-linux-gnu-objdump -t user/_grind | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/grind.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/wc.o user/wc.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_wc user/wc.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_wc > user/wc.asm
riscv64-linux-gnu-objdump -t user/_wc | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/wc.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/zombie.o user/zombie.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_zombie user/zombie.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_zombie > user/zombie.asm
riscv64-linux-gnu-objdump -t user/_zombie | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/zombie.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/uniq.o user/uniq.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_uniq user/uniq.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_uniq > user/uniq.asm
riscv64-linux-gnu-objdump -t user/_uniq | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/uniq.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/head.o user/head.c
riscv64-linux-gnu-ld -z max-page-size=4096 -T user/user.ld -o user/_head user/head.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_head > user/head.asm
riscv64-linux-gnu-objdump -t user/_head | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > user/head.sym
mkfs/mkfs fs.img README user/_cat user/_echo user/_forktest user/_grep user/_init user/_kill user/_ln user/_ls user/_mkdir user/_rm user/_sh user/_stressfs user/_usertests user/_grind user/_wc user/_zombie user/_uniq user/_head
nmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 1954 total 2000
balloc: first 823 blocks have been allocated
balloc: write bitmap block at sector 45
mkfs/mkfs fs.img README uniqfile.txt user/_cat user/_echo user/_forktest user/_grep user/_init user/_kill user/_ln user/_ls user/_mkdir user/_rm user/_sh user/_stressfs user/_usertests user/_grind user/_wc user/_zombie user/_uniq user/_head
nmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 1954 total 2000
balloc: first 824 blocks have been allocated
balloc: write bitmap block at sector 45
mkfs/mkfs fs.img README headfile.txt user/_cat user/_echo user/_forktest user/_grep user/_init user/_kill user/_ln user/_ls user/_mkdir user/_rm user/_sh user/_stressfs user/_usertests user/_grind user/_wc user/_zombie user/_uniq user/_head
nmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 1954 total 2000
balloc: first 824 blocks have been allocated
balloc: write bitmap block at sector 45
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
xv6 kernel is booting
hart 2 starting
hart 1 starting
init: starting sh
$

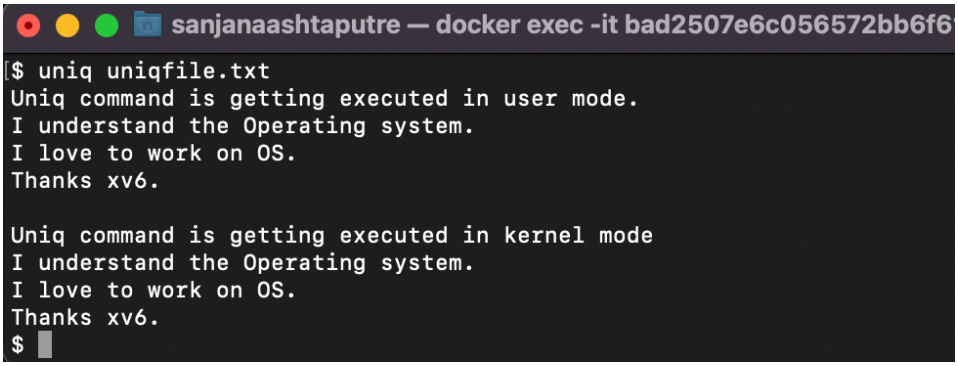
Input files and commands (user and kernel space):

Input file: **uniqfile.txt**



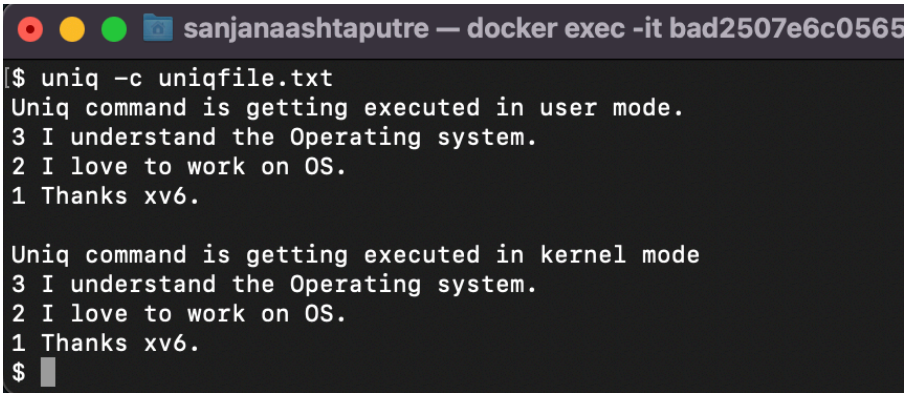
1. **uniq uniqfile.txt**

   Prints by removing the adjacent duplicate lines



2. **uniq -c uniqfile.txt**

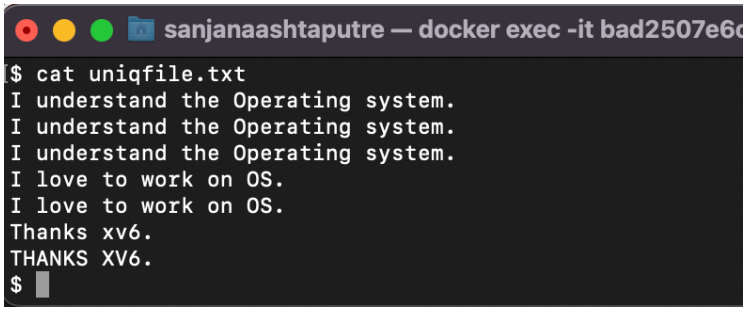
   Prints the number of times a line is repeated before the line.



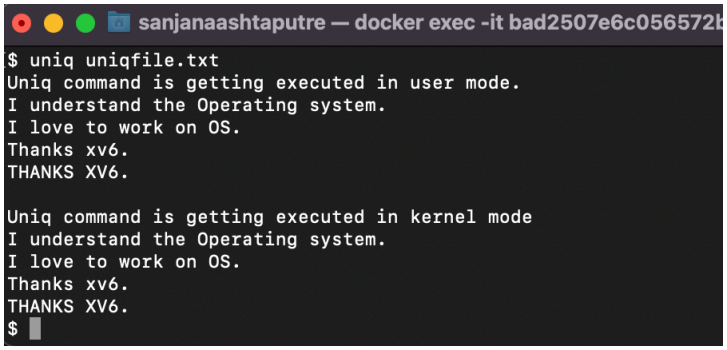
3. **uniq -i uniqfile.txt**

It acts like case insensitive comparison. It removes adjacent duplicate lines and print the lines ignoring the case (upper or lower case is considered as equal)

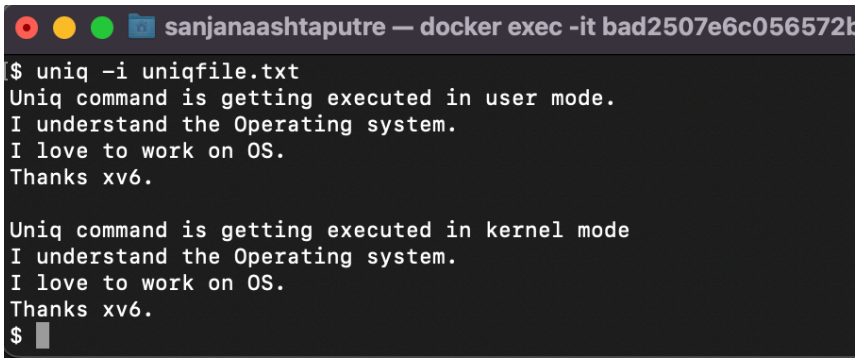
Input file: uniqfile.txt (A line is appended in uppercase)



Without -i command: **uniq uniqfile.txt**



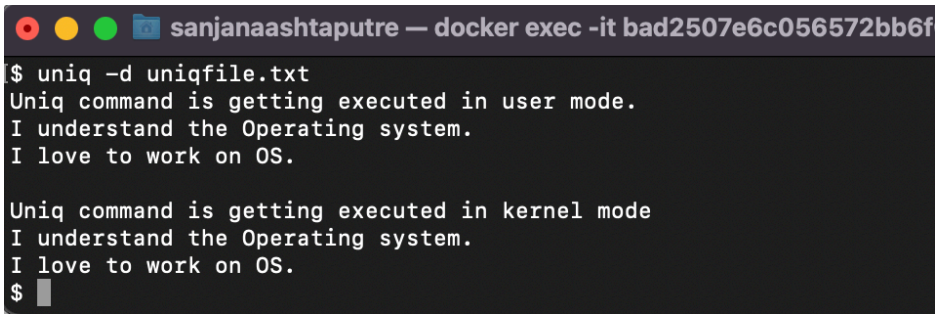
With -i command: **uniq -I uniqfile.txt**



As seen, it is not case sensitive, the last two lines are treated assame.

4. **uniq -d uniqfile.txt**

   Prints only lines that are repeating.



5. **cat uniqfile.txt | uniq**