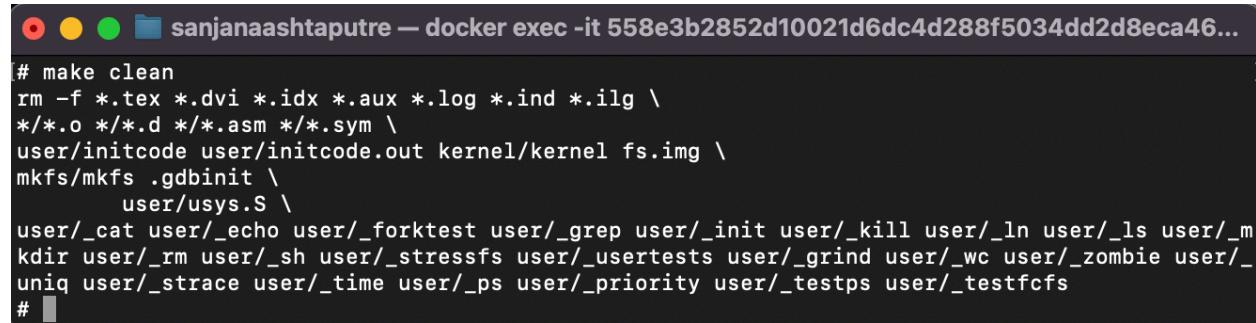


# First-come first-serve (FCFS)

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

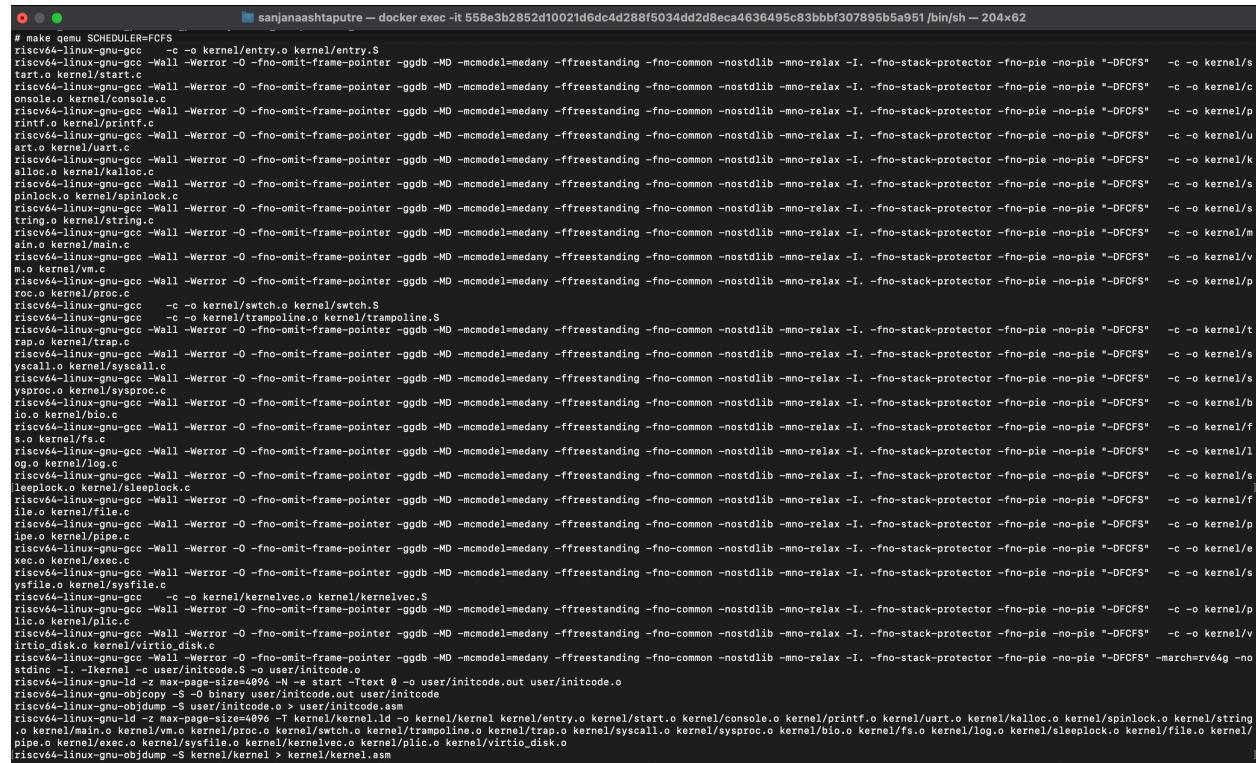
Steps to run the code and its output:

make clean



```
# make clean
rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
*/*.o */*.d */*.asm */*.sym \
user/initcode user/initcode.out kernel/kernel fs.img \
mkfs/mkfs .gdbinit \
user/usys.S \
user/_cat user/_echo user/_forktest user/_grep user/_init user/_kill user/_ln user/_ls user/_m
kdir user/_rm user/_sh user/_stressfs user/_usertests user/_grind user/_wc user/_zombie user/_u
niq user/_strace user/_time user/_ps user/_priority user/_testps user/_testfcfs
#
```

make qemu SCHEDULER=FCFS



```
# make qemu SCHEDULER=FCFS
riscv64-linux-gnu-gcc -c -o kernel/entry.o kernel/entry.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/start.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/console.o kernel/console.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/printf.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/uart.o kernel/uart.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/kalloc.o kernel/kalloc.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/printk.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/string.o kernel/string.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/min.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/vm.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/palloc.o kernel/palloc.c
riscv64-linux-gnu-gcc -c -o kernel/swtch.o kernel/swtch.S
riscv64-linux-gnu-gcc -c -o kernel/trampoline.o kernel/trampoline.S
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/ttrap.o kernel/ttrap.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/syscall.o kernel/syscall.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/sysproc.o kernel/sysproc.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/bio.o kernel/bio.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/fs.o kernel/fs.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/iop.o kernel/iop.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/sleeplock.o kernel/sleeplock.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/file.o kernel/file.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/pipe.o kernel/pipe.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/exec.o kernel/exec.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -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 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/plic.o kernel/plic.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/virtio_disk.o kernel/virtio_disk.c
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o kernel/stdin.o kernel/stdin.c
riscv64-linux-gnu-gcc -N -e start -Ttext 0 -o user/initcode.o 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-objdump -z max-page-size=4096 -T kernel/kernel_id.o kernel/kernel/entry.o kernel/start.o kernel/console.o kernel/printf.o kernel/kalloc.o kernel/spinlock.o kernel/string.o kernel/syscall.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/pine.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
```

```

sanjanaashtaputre — docker exec -it 558e3b2852d10021d6dc4d288f5034dd2d8eca4636495c83bbb3f07895b5a951/bin/sh - 204x62
riscv64-linux-gnu-objdump -S user/_wc | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/wc.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/zombie.o user/zombie.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_zombie user/zombie.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_zombie > user/zombie.asm
riscv64-linux-gnu-objdump -t user/_zombie | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/zombie.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/uniq.o user/uniq.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -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 | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/uniq.sym
riscv64-linux-gnu-objdump -t user/_uniq | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/uniq.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/strace.o user/strace.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_strace user/strace.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_strace > user/strace.asm
riscv64-linux-gnu-objdump -t user/_strace | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/strace.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/timer.o user/time.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_time user/time.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_time > user/time.asm
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/ps.o user/ps.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_ps user/ps.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_ps > user/ps.asm
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/priority.o user/priority.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_priority user/priority.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_priority > user/priority.asm
riscv64-linux-gnu-objdump -t user/_priority | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/priority.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/testtsp.o user/testtsp.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_testtsp user/testtsp.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_testtsp > user/testtsp.asm
riscv64-linux-gnu-objdump -t user/_testtsp | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/testtsp.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -MD -mcmdel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie "-DFCFS" -c -o user/testtfcfs.o user/testtfcfs.c
riscv64-linux-gnu-ld -z max-page-size=4096 -N -e main -Ttext 0 -o user/_testtfcfs user/testtfcfs.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_testtfcfs > user/testtfcfs.asm
riscv64-linux-gnu-objdump -t user/_testtfcfs | sed '1;/SYMBOL TABLE/d; s/.*/ /; /*$/d' > user/testtfcfs.sym
mkfs 46 (boot) user/_testtfcfs 774 blocks 30 inode blocks 13, bitmap blocks 1) blocks 954 total 1800
mknod user/_unmk user/_strace user/_time user/_ps user/_priority user/_testtsp user/_testtfcfs
mknod user/_mk user/_rm user/_sh user/_stressfs user/_userstests user/_grind user/_wc user/_zombie user/_uniq user/_strace user/_time user/_ps user/_priority user/_testtsp user/_testtfcfs
mknod user/_ls user/_mkdir user/_kill user/_grep user/_init user/_user/_ln user/_rm user/_sh user/_stressfs user/_userstests user/_grind user/_wc user/_zombie user/_uniq user/_strace user/_time user/_ps user/_priority user/_testtsp user/_testtfcfs
mknod user/_testtfcfs 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 954 total 1800
ballof: first 783 blocks have been allocated
ballof: write bitmap block at sector 45
mkfs/mkfs fs.img 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/_userstests user/_grind user/_wc user/_zombie user/_uniq user/_strace user/_time user/_ps user/_priority user/_testtsp user/_testtfcfs
mknod 46 (boot) user/_testtfcfs 774 blocks 30 inode blocks 13, bitmap blocks 1) blocks 954 total 1800
ballof: first 783 blocks have been allocated
ballof: write bitmap block at sector 45
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -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:

```

sanjanaashtaputre — docker exec -it 558e3b2852d
[$ cat uniqfile.txt
I understand the Operating system.
I understand the Operating system.
I understand the Operating system.
I love to work on OS.
I love to work on OS.
Thanks xv6.
THANKS XV6.
$ 

```

## Output:

<program file> <number\_of\_processes> <Input\_file>

testfcfs 4 uniqfile.txt

```
terminal  Show  Edit  View  Window  Help
$ testfcfs 4 uniqfile.txt

Child PID 4 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.
Thanks xv6.
THANKS XV6.

Process 1 finished
creation time : 157
run time : 18
end time : 175
wait time : 0
Turnaround Time : 18

Child PID 5 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.
Thanks xv6.
THANKS XV6.

Process 2 finished
creation time : 175
run time : 17
end time : 192
wait time : 0
Turnaround Time : 17

Child PID 6 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.
Thanks xv6.
THANKS XV6.

Process 3 finished
creation time : 192
run time : 16
end time : 208
wait time : 0
Turnaround Time : 16

Child PID 7 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.
Thanks xv6.
THANKS XV6.

Process 4 finished
creation time : 208
run time : 16
end time : 224
wait time : 0
Turnaround Time : 16

Average run time 16,  Average wait time 0
$
```

<program file> <number\_of\_processes> <-c/-d/-i> <Input\_file>

**testfcfs 4 -c uniqfile.txt**

```
$ testfcfs 4 -c uniqfile.txt
Child PID 16 created
Uniq command is getting executed in kernel mode
3 I understand the Operating system.
2 I love to work on OS.
1 Thanks xv6.
1 THANKS XV6.

Process 1 finished
creation time : 2106
run time : 16
end time : 2122
wait time : 0
Turnaround Time : 16

Child PID 17 created
Uniq command is getting executed in user mode.
3 I understand the Operating system.
2 I love to work on OS.
1 Thanks xv6.
1 THANKS XV6.

Process 2 finished
creation time : 2122
run time : 17
end time : 2139
wait time : 0
Turnaround Time : 17

Child PID 18 created
Uniq command is getting executed in kernel mode
3 I understand the Operating system.
2 I love to work on OS.
1 Thanks xv6.
1 THANKS XV6.

Process 3 finished
creation time : 2139
run time : 15
end time : 2154
wait time : 0
Turnaround Time : 15

Child PID 19 created
Uniq command is getting executed in user mode.
3 I understand the Operating system.
2 I love to work on OS.
1 Thanks xv6.
1 THANKS XV6.

Process 4 finished
creation time : 2154
run time : 15
end time : 2169
wait time : 0
Turnaround Time : 15

Average run time 15, Average wait time 0
$
```

## testfcfs 5 -i uniqfile.txt

```
sanjanaashtaputre — docker exec -it 558e3b2852d10021d6dc4$ testfcfs 5 -i uniqfile.txt

Child PID 21 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.
Thanks xv6.

Process 1 finished
creation time : 2994
run time : 15
end time : 3009
wait time : 0
Turnaround Time : 15

Child PID 22 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.
Thanks xv6.

Process 2 finished
creation time : 3009
run time : 16
end time : 3025
wait time : 0
Turnaround Time : 16

Child PID 23 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.
Thanks xv6.

Process 3 finished
creation time : 3025
run time : 16
end time : 3041
wait time : 0
Turnaround Time : 16

Child PID 24 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.
Thanks xv6.

Process 4 finished
creation time : 3041
run time : 16
end time : 3058
wait time : 1
Turnaround Time : 17

Child PID 25 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.
Thanks xv6.

Process 5 finished
creation time : 3058
run time : 14
end time : 3072
wait time : 0
Turnaround Time : 14

Average run time 15, Average wait time 0
$
```

**testfcfs 7 -d uniqfile.txt**

```
$ testfcfs 7 -d uniqfile.txt
Child PID 35 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.

Process 1 finished
creation time : 6654
run time : 15
end time : 6669
wait time : 0
Turnaround Time : 15

Child PID 36 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.

Process 2 finished
creation time : 6669
run time : 16
end time : 6685
wait time : 0
Turnaround Time : 16

Child PID 37 created

Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.

Process 3 finished
creation time : 6685
run time : 15
end time : 6700
wait time : 0
Turnaround Time : 15

Child PID 38 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.

Process 4 finished
creation time : 6700
run time : 15
end time : 6715
wait time : 0
Turnaround Time : 15
```

```
Child PID 39 created
Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.

Process 5 finished
creation time : 6715
run time : 17
end time : 6732
wait time : 0
Turnaround Time : 17

Child PID 40 created
Uniq command is getting executed in user mode.
I understand the Operating system.
I love to work on OS.

Process 6 finished
creation time : 6732
run time : 13
end time : 6745
wait time : 0
Turnaround Time : 13

Child PID 41 created
Uniq command is getting executed in kernel mode
I understand the Operating system.
I love to work on OS.

Process 7 finished
creation time : 6745
run time : 15
end time : 6760
wait time : 0
Turnaround Time : 15

Average run time 15, Average wait time 0
$
```

### Code logic:

While running *make qemu*, Scheduler=FCFS should be mentioned. It is added to the Makefile accordingly.

#### File: *testfcfs.c*

This file is created in the user directory of xv6. Here, a ‘p’ new processes are created using *fork()*. A dummy for loop is added just to consume run time with *uniq* function. *Uniq* command is called both in user and kernel mode, once the process is completed, the statistics of the process are displayed. After running all the processes, the average wait and run time is calculated and displayed at the end. *uniq\_f* is the user function containing the *uniq* functionality.

```

/riscv-scheduling/user/testfcfs.c
```

```

123     argv[i]++;
124     var_icd = *argv[i];
125     f=1;
126   }
127   if(f==0){
128     if(n&2 == 0){
129       while((n2 = read(fd, buffer, sizeof(buffer))) > 0 )
130       {
131         uniq(n2, buffer, var_icd);
132       }
133     }
134     if (n2 < 0) {
135       printf("Error: Unable to read file\n");
136     }
137   }else{
138     uniq_f(fd, var_icd);
139   }
140   close(fd);
141   f=0;
142 }
143
144 close(fd);
145 printf("\nProcess %d finished", n+1);
146 exit(0);
147 } else {
148   if(waitx(0,&rt,&wt) >= 0) {
149     rt += rt;
150     twt += wt;
151   }
152 }
153
154 printf("\nAverage run time %d, Average wait time %d\n", rt / p, twt / p);
155 exit(0);
156 }
```

## File: **defs.h**

The functions are declared here like

```
Int updateTime(void);
```

## File: **proc.c**

In allocproc(), all the other process fields are initialized.

In exit(), the process end time is obtained.

In scheduler(), the changes are made to the default round robin scheduling as shown below. Here, the process that has arrived early, I,e. Which has least creation time is picked from RUNNABLE state and made to run with RUNNING state.

```

/xv6-riscv-scheduling/kernel/proc.c
```

```

505 #ifdef FCFS
506 struct proc* fp = 0;
507
508 for (p = proc; p < &proc[NPROC]; p++)
509 {
510   acquire(&p->lock);
511   if ([p->state == RUNNABLE])
512   {
513     if (!fp || p->timeOfCreation < fp->timeOfCreation)
514     {
515       if (fp)
516         release(&fp->lock);
517       fp = p;
518       continue;
519     }
520   }
521   release(&p->lock);
522 }
```

```

523
524   }
525
526   if (fp)
527   {
528     fp->state = RUNNING;
529
530     c->proc = fp;
531     p->numScheduled++;
532     swtch(&c->context, &fp->context);
533
534     c->proc = 0;
535     release(&fp->lock);
536   }
537
538 #endif
539 #endif
540 }
541 }
```

updateTime() function is added to the file where the run time and sleep time of the process is updated.

```
/xv6-riscv-scheduling/kernel/proc.c

756 void
757 updateTime()
758 {
759     struct proc *p;
760     for (p = proc; p < &proc[NPROC]; p++)
761     {
762         acquire(&p->lock);
763
764         if (p->state == RUNNING)
765         {
766             p->runTime++;
767             p->totalRunTime++;
768         }
769
770         if (p->state == SLEEPING)
771             p->sleepTime++;
772
773         release(&p->lock);
774 }
```

waitx() function is added to the file, where total run time, turnaround time, wait time are calculated.

#### File: **proc.h**

All the process fields are declared here. Fields for run time, end time, scheduling time, etc.

#### File: **syscall.c**

All the function definitions are declared here.

#### File: **syscall.h**

All the system call numbers for the system calls are assigned here.

#### File: **sysproc.c**

All system call inputs are passed here for the respective functionality.

#### File: **trap.c**

The yield() is blocked here in order stop the process from preemption on the clock interrupt as by default xv6 follows round robin scheduling.

#### File: **user.h**

The system call function is declared.

#### File: **usys.pl**

The system call entry is declared like: entry("uniq");

