

## Implementation of head command

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

### Code logic:

File: head.c

All the variables are declared. If no filename is given, it is read as a standard input and passed to

```
/xv6-riscv/user/head.c

```
36 v int main(int argc, char *argv[]) {
37     int i,n;
38     int fd = 0;
39     int fdk = 0;
40     int flines = 14;
41     char *file;
42     char a;
43     int f = 0;
44     char buffer[512];
45     file = "";
46
47 v     if (argc <= 1) {
48         head_f(0, "", flines);
49         while((n = read(0, buffer, sizeof(buffer))) > 0 )
50 v     {
51         head(0, buffer, flines);
52     }
53 v     if (n < 0) {
54         printf("Error: Unable to read file(Kernel mode)\n");
55     }
56     exit(0);
57 }
```

the function and the system call. An appropriate error message is printed on failure.

```
/xv6-riscv/user/head.c

C

57     }
58 v     else {
59 v         for (i = 1; i < argc; i++) {
60             a = *argv[i];
61             if(a != '-'){
62                 if ((fd = open(argv[i], 0)) < 0){
63                     printf("Error: Cannot Open File %s\n", argv[i]);
64                     exit(1);
65                 }
66             }
67             if (a == '-') {
68                 argv[i]++;
69                 flines = atoi(argv[i]++);
70                 f = 1;
71 }
```

If ‘-‘ number of lines is given, it is stored in a variable *flines* and a flag *f* is set.

The arguments are passed to the function *head\_f* where the user mode implementation is coded. The file is read and the content is passed to the *head()* system call and an appropriate error message is displayed if any. If multiple files are given, each file content is displayed preceded with the file name.

```
/xv6-riscv/user/head.c
```

```
72 v         if(f == 0){
73             printf("\nFile: %s\n", argv[i]);
74             head_f(fd, file, flines);
75             fdk = open(argv[i], 0);
76             while((n = read(fd, buffer, sizeof(buffer))) > 0 )
77             {
78                 printf("\nFile: %s\n", argv[i]);
79                 head(n, buffer, flines);
80             }
81 v         if (n < 0) {
82             printf("Error: Unable to read file(kernel mode)\n");
83             close(fd);
84         }
85     }
86     f = 0;
87 }
88     close(fd);
89     exit(0);
90 }
```

Each character of the file is read and printed until a line is completed.

```
/xv6-riscv/user/head.c
```

```
1 #include "kernel/types.h"
2 #include "kernel/stat.h"
3 #include "user.h"
4
5 char buf[512];
6
7 int head_f(int fd, char *name, int line)
8 {
9     int i, n, l_no;
10    l_no = 0;
11    printf("Head command is getting executed in user mode\n");
12    while((n = read(fd, buf, sizeof(buf))) > 0 )
13    {
14        for(i=0;i<=n ;i++)
15        {
16            if(buf[i]!='\n'){
17                printf("%c",buf[i]);
18            }
19        }
20    }
21    close(fd);
22    exit(0);
23 }
```

*l\_no* variable contains the total line number iterated. If it is equal to the provided line number, it returns back to the main block else the next line is read. If line number is not given, it is defaulted to 14 in main block which is passed to this function.

```
/xv6-riscv/user/head.c

18      }
19 v    else if (l_no == (line-1)){
20        printf("\n");
21        return 0;
22    }
23 v    else{
24        printf("\n");
25        l_no++;
26    }
27 }
28 }
29 v if(n < 0){
30     printf("head: unable to read(user mode)\n");
31     exit(1);
32 }
33 return 0;
34 }
35
```

File: sysproc.c

```
/xv6-riscv/kernel/sysproc.c
```

C

```
201
202 uint64
203 sys_head(void)
204 v {
205     int n,i;
206     int l=0;
207     char buffer[512];
208     int line;
209     char convStr[2];
210     argint(0, &n);
211     argstr(1, buffer, sizeof(buffer));
212     argint(2, &line); //lines to print - default 14
213     printf("Head command is getting executed in kernel mode\n");
214     for(i=0;i<=n ;i++)
215 v     {
216 v         if(buffer[i]!='\n'){
217             convStr[0] = buffer[i];
218             convStr[1]='\0';
219             printf("%s", convStr);
220         }
221 }
```

The arguments are read into respective variables. The file content is iterated over and each character is printed until the end of the line. As ‘%c’ is not allowed, char is stored in a string and printed.

```

21 v      else if (l == (line-1)){
22         printf("\n");
23         return 0;
24     }
25 v     else{
26         printf("\n");
27         l++;
28     }
29 }
30     return 0;
31 }
```

The total lines iterated is stored in a variable, once it reaches the given line number it is returned back to the main program. If the line number is not given, it is defaulted to 14.

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
```

```
/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);
```

```
/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,
133 };
134
```

syscall.h

syscall.c

usys.pl

```
/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");
```

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);
```

## Input file and the 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/_mkdir \
user/_rm user/_sh user/_stressfs user/_usertests user/_grind user/_wc user/_zombie user/_uniq user_ \
_head
#
```

Make qemu

```
# make qemu
riscv64-linux-gnu-gcc -c -o kernel/entry.o kernel/entry.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/
start.o kernel/start.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/
console.o kernel/console.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/
printf.o kernel/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 kernel/
uart.o kernel/uart.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/
kalloc.o kernel/kalloc.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/
spinlock.o kernel/spinlock.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/
string.o kernel/string.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/
main.o kernel/main.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/
vm.o kernel/vm.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/
proc.o kernel/proc.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 -gdwarf-2 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/
trap.o kernel/trap.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/
syscall.o kernel/syscall.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/
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/
```

```

● ● ● sanjanaashtaputre — docker exec -it bad2507e6c056572bb6f61188947ad36746178abcf1e15e0cc5b411967489d53 /bin/sh — 204x62
riscv64-linux-gnu-objdump -t user/_sh | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/sh.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/stressfs.o user/stressfs.c
riscv64-linux-gnu-objdump -S user/_stressfs > user/stressfs.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -t user/_stressfs | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/stressfs.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mmodel=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-objdump -z max-page-size=4096 -T user/user_id -o user/_user_id.o user/stressfs.o user/uib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_user_id > user/_user_id.sym
riscv64-linux-gnu-objdump -t user/_user_id | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_user_id.sym
riscv64-linux-gnu-objdump -S user/_grind > user/_grind.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -t user/_grind | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_grind.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mmodel=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-objdump -t user/_usertests | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_usertests.sym
riscv64-linux-gnu-objdump -S user/_wc > user/_wc.o
riscv64-linux-gnu-objdump -t user/_wc | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_wc.sym
riscv64-linux-gnu-objdump -S user/_zombie > user/_zombie.o
riscv64-linux-gnu-objdump -t user/_zombie | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_zombie.sym
riscv64-linux-gnu-objdump -S user/_zombie > user/_zombie.o user/uib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -t user/_zombie | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_zombie.o
riscv64-linux-gnu-objdump -S user/_uniq > user/_uniq.o
riscv64-linux-gnu-objdump -t user/_uniq | sed '1,/SYMBOL TABLE/d; s/.* /;/;$d' > user/_uniq.sym
riscv64-linux-gnu-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-stack-protector -fno-pie -no-pie -c -o user/head.ad.o user/head.c
riscv64-linux-gnu-objdump -z max-page-size=4096 -T user/_user_id -o user/_uniq user/uib.o user/usys.o user/printf.o user/umalloc.o
riscv64-linux-gnu-objdump -S user/_head > user/_head.o
riscv64-linux-gnu-objdump -t user/_head | sed '1,/SYMBOL TABLE/d; s/.* /;/;$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 839 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 820 blocks have been allocated
balloc: write bitmap block at sector 45
nmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 1954 total 2000
balloc: first 820 blocks have been allocated
balloc: write bitmap block at sector 45
genuv-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,drives=x0,bus=virtio-mmio-bus.0
xv6 kernel is booting
hart 1 starting
hart 2 starting
init: starting sh
$ 

```

## 1. head headfile.txt

```

● ● ● sanjanaashtaputre — docker exec -it bad2507e6c056572bb6f611889
$ head headfile.txt

File: headfile.txt
Head command is getting executed in user mode
This is sentence 1.
This is sentence 2.
This is sentence 3.
This is sentence 4.
This is sentence 5.
This is sentence 6.
This is sentence 7.
This is sentence 8.
This is sentence 9.
This is sentence 10.
This is sentence 11.
This is sentence 12.
This is sentence 13.
This is sentence 14.

File: headfile.txt
Head command is getting executed in kernel mode
This is sentence 1.
This is sentence 2.
This is sentence 3.
This is sentence 4.
This is sentence 5.
This is sentence 6.
This is sentence 7.
This is sentence 8.
This is sentence 9.
This is sentence 10.
This is sentence 11.
This is sentence 12.
This is sentence 13.
This is sentence 14.
$ 

```

```
[sanjanaashtaputre — docker exec -it bad2507e6c05657]$ head -7 headfile.txt  
File: headfile.txt  
Head command is getting executed in user mode  
This is sentence 1.  
This is sentence 2.  
This is sentence 3.  
This is sentence 4.  
This is sentence 5.  
This is sentence 6.  
This is sentence 7.  
  
File: headfile.txt  
Head command is getting executed in kernel mode  
This is sentence 1.  
This is sentence 2.  
This is sentence 3.  
This is sentence 4.  
This is sentence 5.  
This is sentence 6.  
This is sentence 7.  
$
```

## 2. head -7 headfile.txt

Top 7 lines of the files are printed

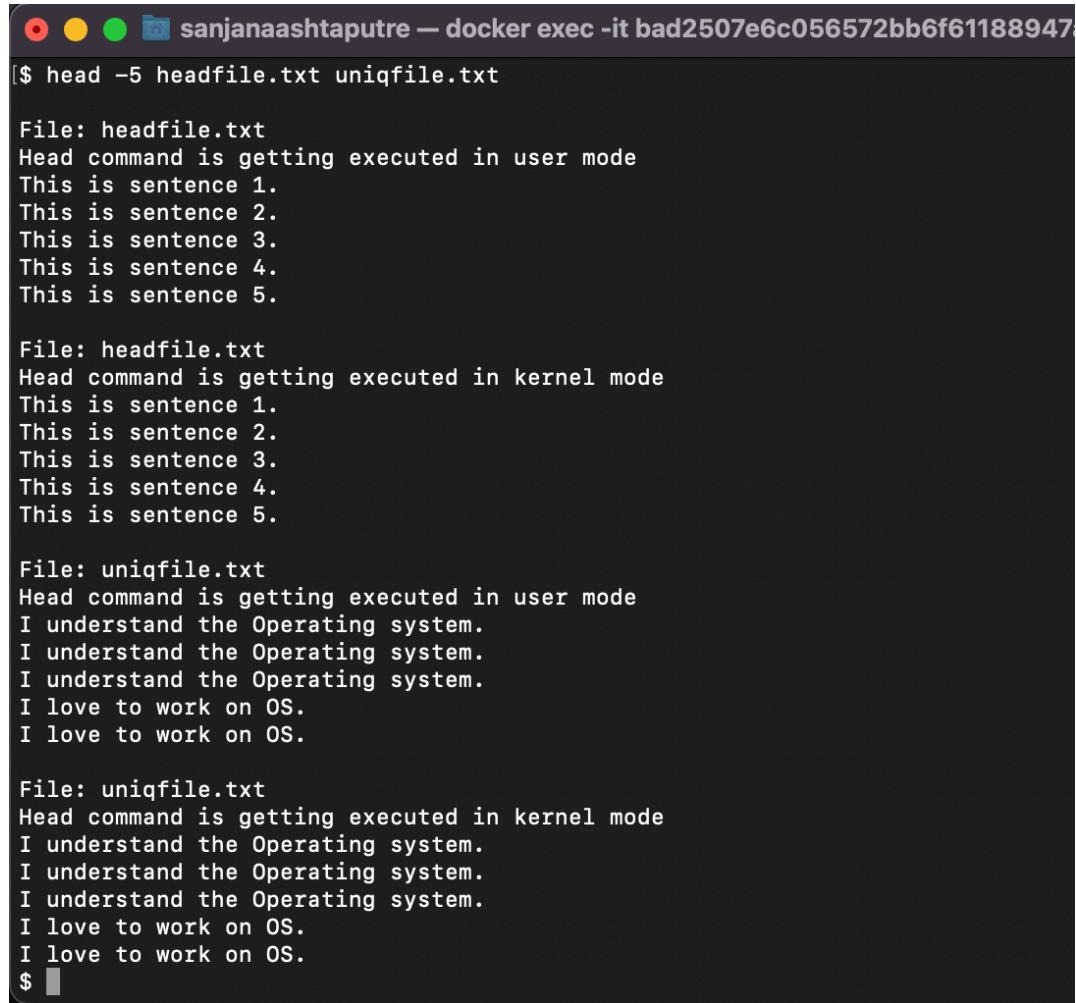
```
[sanjanaashtaputre — docker exec -it bad2507e6c056572bb6f]$ head uniqfile.txt headfile.txt  
File: uniqfile.txt  
Head command is getting executed in user mode  
I understand the Operating system.  
I understand the Operating system.  
I understand the Operating system.  
I love to work on OS.  
I love to work on OS.  
Thanks xv6  
  
File: uniqfile.txt  
Head command is getting executed in kernel mode  
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  
  
File: headfile.txt  
Head command is getting executed in user mode  
This is sentence 1.  
This is sentence 2.  
This is sentence 3.  
This is sentence 4.  
This is sentence 5.  
This is sentence 6.  
This is sentence 7.  
This is sentence 8.  
This is sentence 9.  
This is sentence 10.  
This is sentence 11.  
This is sentence 12.  
This is sentence 13.  
This is sentence 14.  
  
File: headfile.txt  
Head command is getting executed in kernel mode  
This is sentence 1.  
This is sentence 2.  
This is sentence 3.  
This is sentence 4.  
This is sentence 5.  
This is sentence 6.  
This is sentence 7.  
This is sentence 8.  
This is sentence 9.  
This is sentence 10.  
This is sentence 11.  
This is sentence 12.  
This is sentence 13.  
This is sentence 14.  
$
```

## 3. head uniqfile.txt head file.txt

When more than one file is given, both the contents upto 14 (default) lines are printed preceded with the file names.

#### 4. head -5 uniqfile.txt head file.txt

When more than one file is given, both the contents upto 5 lines are printed preceded with the file names.



```
$ head -5 headfile.txt uniqfile.txt

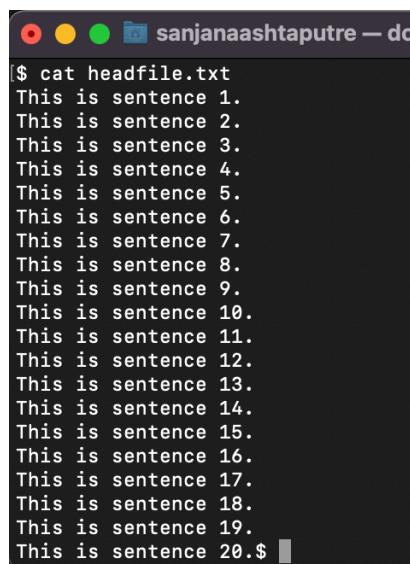
File: headfile.txt
Head command is getting executed in user mode
This is sentence 1.
This is sentence 2.
This is sentence 3.
This is sentence 4.
This is sentence 5.

File: headfile.txt
Head command is getting executed in kernel mode
This is sentence 1.
This is sentence 2.
This is sentence 3.
This is sentence 4.
This is sentence 5.

File: uniqfile.txt
Head command is getting executed in user mode
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.

File: uniqfile.txt
Head command is getting executed in kernel mode
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.
$
```

Input **headfile.txt**



```
$ cat headfile.txt
This is sentence 1.
This is sentence 2.
This is sentence 3.
This is sentence 4.
This is sentence 5.
This is sentence 6.
This is sentence 7.
This is sentence 8.
This is sentence 9.
This is sentence 10.
This is sentence 11.
This is sentence 12.
This is sentence 13.
This is sentence 14.
This is sentence 15.
This is sentence 16.
This is sentence 17.
This is sentence 18.
This is sentence 19.
This is sentence 20.$
```