Lab Assignment 2: Hands-on with xv6 OS

Roll No: CS22BTECH11050
Name : Rishitha Surineni

Task-1.1:

Methodology:

In the file sleep.c

Check the number of arguments passed by the user.

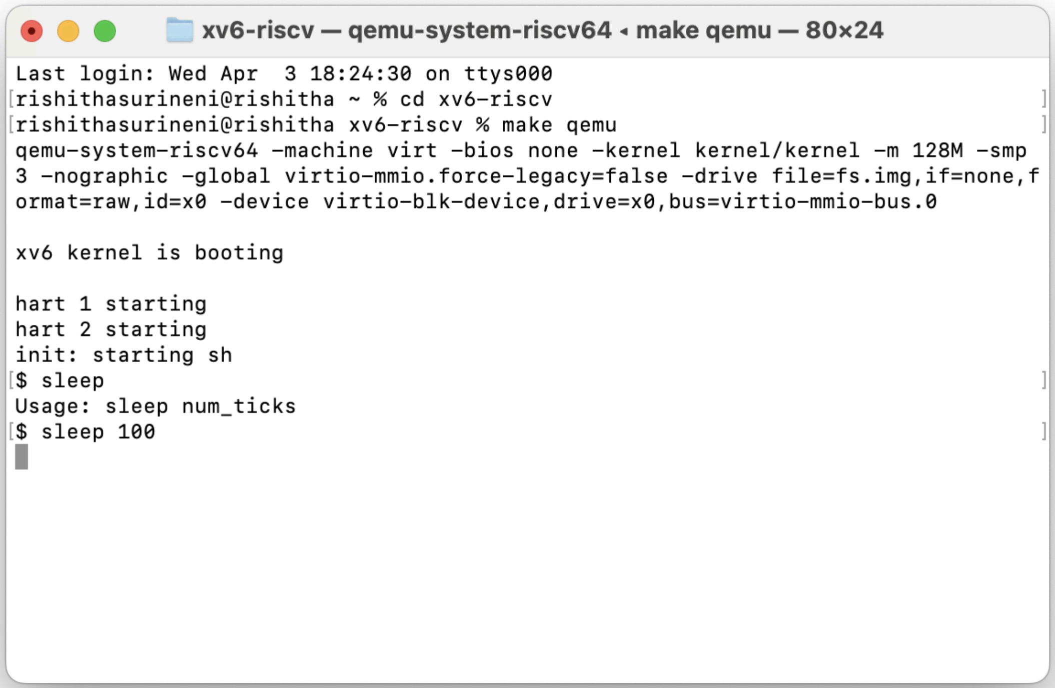
If this is less than 2 then display an error message.

Convert the second argument i.e argv[1] into integer using atoi() function and store it in a variable named n.

Call the function sleep() with n as parameter.

Then the program will sleep for n clockticks and then the program is exited by exit(0).

Output:



```
Last login: Wed Apr  3 18:24:30 on ttys000
rishithasurineni@rishitha ~ % cd xv6-riscv
rishithasurineni@rishitha xv6-riscv % make qemu
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,f
ormat=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$ sleep
Usage: sleep num_ticks
$ sleep 100
```

In the output screenshot it can be seen that the program sleeps for 100 clock ticks so the shell waits for 100 clock ticks and then the program is exited.

This program is added to the makefile and then it can be executed by running
make quem and then
sleep <number of clock ticks>

Task-1.2:

Methodology:

In the file pingpong.c

Two pipes are created for communication between parent and child.
The descriptors of these pipes are stored in integer arrays.
Child process is created using fork() system call.
fork() return 0 for child process and PID of child for parent process.

In child process

The writing end of ParentPipe is closed using close() and then the contents of the pipe are read using read() and it is stored in a character array.

The output message is printed out and this contains the PID of the child process.

Then the reading end of this pipe is closed.

The reading end of ChildPipe is closed and then a character is written into the pipe using write() and then this writing end is closed.

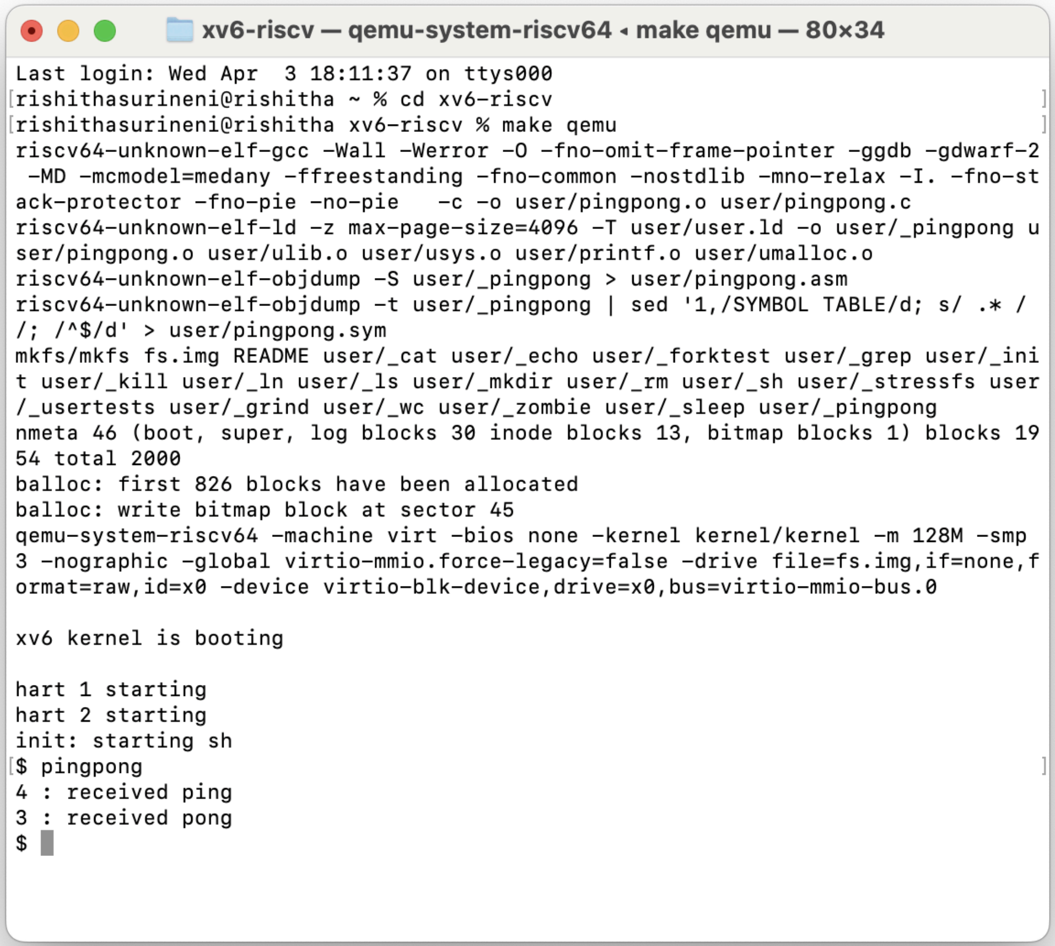
In parent process

The reading end of Parent pipe is closed and then a character is written into the parent pipe using write(). Then the writing end of this pipe is closed.

Then the parent waits till the child process is completed using wait(NULL) (in this time the child finishes writing into the childpipe).

The writing end of childpipe is closed and the contents of it are read using read(). The output message is printed with PID of parent. And then this reading end is closed.

Output:



```
Last login: Wed Apr  3 18:11:37 on ttys000
rishithasurineni@rishitha ~ % cd xv6-riscv
rishithasurineni@rishitha xv6-riscv % make qemu
riscv64-unknown-elf-gcc -Wall -Werror -O -fno-omit-frame-pointer -ggdb -gdwarf-2
 -MD -mcmodel=medany -ffreestanding -fno-common -nostdlib -mno-relax -I. -fno-st
ack-protector -fno-pie -no-pie   -c -o user/pingpong.o user/pingpong.c
riscv64-unknown-elf-ld -z max-page-size=4096 -T user/user.ld -o user/_pingpong u
ser/pingpong.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-unknown-elf-objdump -S user/_pingpong > user/pingpong.asm
riscv64-unknown-elf-objdump -t user/_pingpong | sed '1,/SYMBOL TABLE/d; s/ .* /
/; /^$/d' > user/pingpong.sym
mkfs/mkfs fs.img README user/_cat user/_echo user/_forktest user/_grep user/_ini
t user/_kill user/_ln user/_ls user/_mkdir user/_rm user/_sh user/_stressfs user
/_usertests user/_grind user/_wc user/_zombie user/_sleep user/_pingpong
nmeta 46 (boot, super, log blocks 30 inode blocks 13, bitmap blocks 1) blocks 19
54 total 2000
balloc: first 826 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,f
ormat=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$ pingpong
4 : received ping
3 : received pong
$ 
```