

Assignment 3

Neeraj Pandey, Raunak Basu

Structure of the file:

1. NeerajPandey_CS1217_Assignment3.pdf contains the screenshots of output and short comments.
2. Folder “question-1” includes the syscall.c file which has been updated for Question 1 as per the requirements of the assignment
3. Folder “question-2” includes the files which had been updated for Question 2 as per the requirements of the assignment

0. Boot xv6

xv6 was booted by running the commands as shown in the screenshots below.

```
~/gdbinit (Downloads) - Sublime Text (UNREGISTERED)
~/Desktop/os/assignment_3/xv6-public master make qemu-nox-gdb
*** Now run 'gdb'.
qemu-system-i386 -nographic -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512 -S -gdb tcp::26000
```

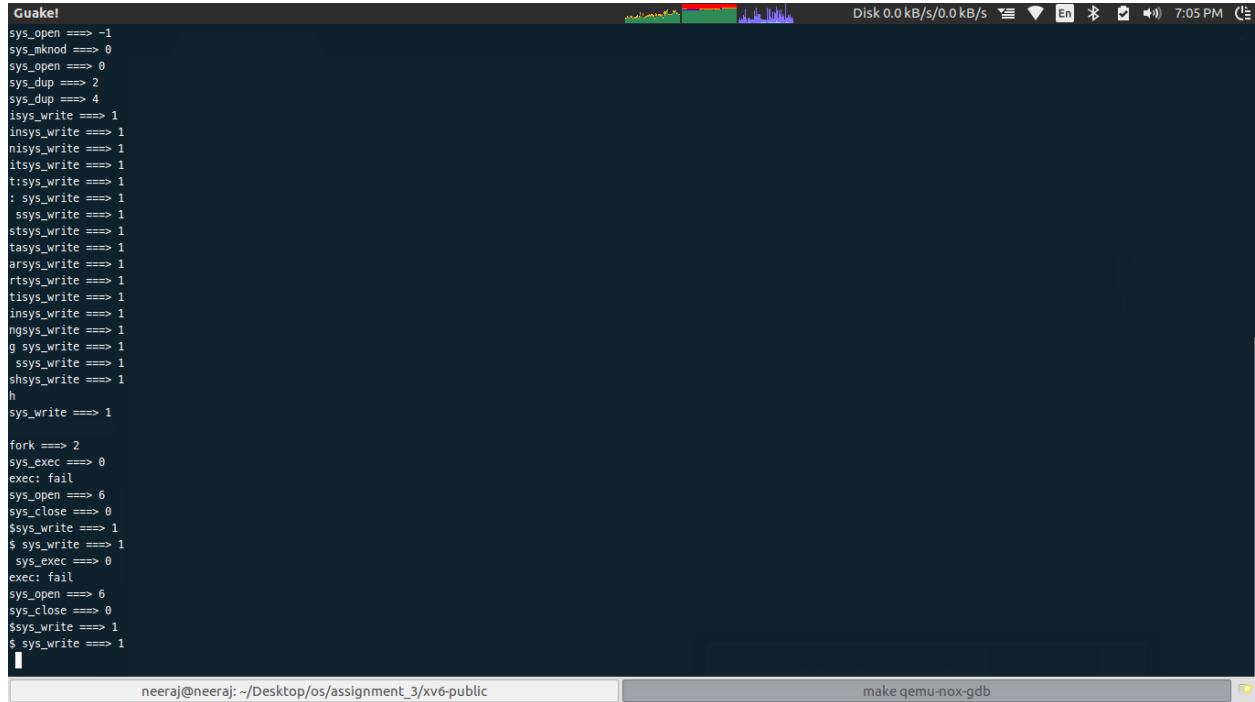
```
Guake!
~/Desktop/os/assignment_3/xv6-public
~/Desktop/os/assignment_3/xv6-public master gdb
GNU gdb (Ubuntu 7.11.1-0ubuntu16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
+ target remote localhost:26000
warning: A handler for the OS ABI "GNU/Linux" is not built into this configuration
of GDB. Attempting to continue with the default i386 settings.

The target architecture is assumed to be i386
{f000:fff0} 0xffff0: jmp $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file kernel
(gdb) br * add-auto-load-safe-path /home/neeraj/Desktop/os/assignment_3/xv6-public/.gdbinit
No symbol "add" in current context.
(gdb) br * 0x0010000c
Breakpoint 1 at 0x10000c
(gdb) c
Continuing.
The target architecture is assumed to be i386
=> 0x10000c: mov %cr4,%eax

Thread 1 hit Breakpoint 1, 0x0010000c in ?? ()
(gdb)
```

1. System call tracing

The xv6 kernel was modified to print out a line for each system call invocation. The required output was found, as shown in screenshot below.



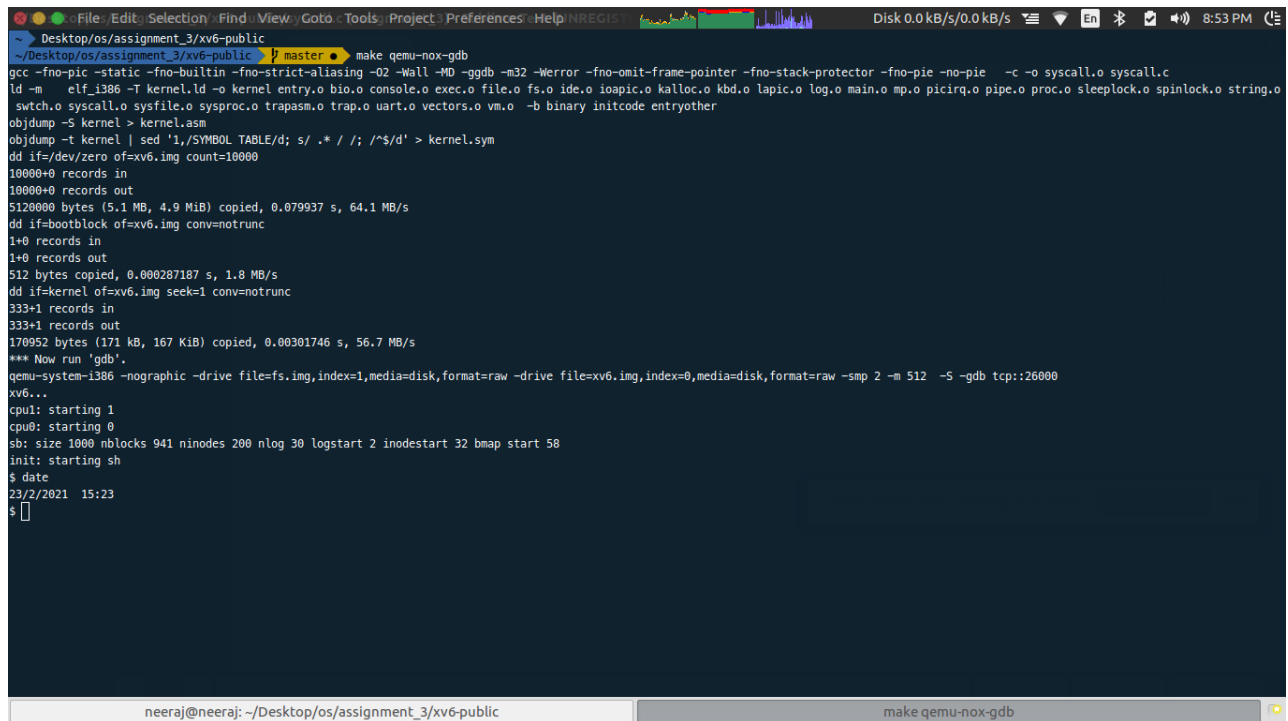
```
Guake
sys_open ==> -1
sys_mknod ==> 0
sys_open ==> 0
sys_dup ==> 2
sys_dup ==> 4
isys_write ==> 1
insys_write ==> 1
nisisys_write ==> 1
itsys_write ==> 1
t:sys_write ==> 1
: sys_write ==> 1
. sys_write ==> 1
stsys_write ==> 1
tasys_write ==> 1
arsys_write ==> 1
rtsys_write ==> 1
t:sys_write ==> 1
insys_write ==> 1
ngsys_write ==> 1
g sys_write ==> 1
. sys_write ==> 1
shsys_write ==> 1
h
sys_write ==> 1

fork ==> 2
sys_exec ==> 0
exec: fail
sys_open ==> 6
sys_close ==> 0
ssys_write ==> 1
$ sys_write ==> 1
. sys_exec ==> 0
exec: fail
sys_open ==> 6
sys_close ==> 0
ssys_write ==> 1
$ sys_write ==> 1
```

neeraj@neeraj: ~/Desktop/os/assignment_3/xv6-public make qemu-nox-gdb

2. Date System Call

- syscall.h: We update syscall.h and define date function with an associated value
- syscall.c: Update the external variables and syscall array with the above function
- user.h: Add the struct argument with a pointer to the cmostime() function in lapic.c
- date.c: Print date and time in the format DD/MM/YYYY MM:HH
- usys.s: We add "SYSCALL(date)"
- Update the Makefile with "_date"
- sysproc.c: Create system call function for date in order to generate a new system call
- Typing date in the xv6 shell gives the current UTC time: (result: 23/2/2021 15:23)



```
Desktop/os/assignment_3/xv6-public
~/Desktop/os/assignment_3/xv6-public master make qemu-nox-gdb
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie -c -o syscall.o syscall.c
ld -m elf_i386 -T kernel.ld -o kernel entry.o bio.o console.o exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o spinlock.o string.o
switch.o syscall.o sysfile.o sysproc.o trapasm.o trap.o uart.o vectors.o vm.o -b binary initcode entryother
objdump -S kernel > kernel.asm
objdump -t kernel | sed '1,/SYMBOL TABLE/d; s/ .* / /; /$/d' > kernel.sym
dd if=/dev/zero of=xv6.img count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.079937 s, 64.1 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 records in
1+0 records out
512 bytes copied, 0.000207107 s, 1.8 MB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
333+1 records in
333+1 records out
170952 bytes (171 kB, 167 KiB) copied, 0.00301746 s, 56.7 MB/s
*** Now run 'gdb'.
qemu-system-i386 -nographic -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512 -S -gdb tcp::26000
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ date
23/2/2021 15:23
$
```