Name: Ken Lin

SID: 006198682

Github repository of code: https://github.com/DehNutCase/CSE-461/tree/master/lab1

Source code of simple shell (SimpleShell.c in my github repository):

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>
#include <string.h>

void read_command (char cmd[], char *par[])
{
    char line[1024];
    int count = 0, i = 0, j = 0;
    char *array[100], *pch;

    while(1){
        int c = fgetc (stdin);
        line[count++] = (char) c;
        if (c == '\n') break;
    }

    if (count == 1)return;
    pch = strtok (line, "\n");

    while (pch != NULL){
        array[i++] = strdup(pch);
```

```
        pch = strtok(NULL, "\n");
    }

    strcpy(cmd, array[0]);

    for (int j = 0; j < i; j++)
        par[j] = array[j];
    par[i] = NULL;

    return;
}

void type_prompt()
{
    static int first_time = 1;
    if (first_time){
        const char* CLEAR_SCREEN_ANSI = "\e[1;1H\e[2J";
        //write(STDOUT_FILENO,CLEAR_SCREEN_ANSI, 12);
        printf(CLEAR_SCREEN_ANSI);
        first_time = 0;
    }

    printf("#"); //# is prompt
}


int main()
{
```

```
    char cmd[100], command[100], *parameters[20];
    //enviroment variable
    char *envp[] = { (char *) "PATH=/bin", 0};
    while (1){
        type_prompt(); //display prompt on screen
        read_command ( command, parameters ); //read terminal inputs
        if (fork() != 0)
            wait (NULL);
        else{
            strcpy (cmd, "/bin/");
            strcat (cmd, command);
            execve (cmd, parameters, envp);
        }
        if ( strcmp (command, "exit") == 0 )
            break;
    }

    return 0;
}
```

Selected Files from xv6 folder:

Makefile:

```
OBJS = \
        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\
	swtch.o\
	syscall.o\
	sysfile.o\
	sysproc.o\
	trapasm.o\
	trap.o\
	uart.o\
	vectors.o\
	vm.o\

# Cross-compiling (e.g., on Mac OS X)
# TOOLPREFIX = i386-jos-elf

# Using native tools (e.g., on X86 Linux)
```

```
#TOOLPREFIX =

# Try to infer the correct TOOLPREFIX if not set
ifndef TOOLPREFIX
TOOLPREFIX := $(shell if i386-jos-elf-objdump -i 2>&1 | grep '^elf32-i386$$' >/dev/null 2>&1; \
	then echo 'i386-jos-elf-'; \
	elif objdump -i 2>&1 | grep 'elf32-i386' >/dev/null 2>&1; \
	then echo ''; \
	else echo "***" 1>&2; \
	echo "*** Error: Couldn't find an i386-*-elf version of GCC/binutils." 1>&2; \
	echo "*** Is the directory with i386-jos-elf-gcc in your PATH?" 1>&2; \
	echo "*** If your i386-*-elf toolchain is installed with a command" 1>&2; \
	echo "*** prefix other than 'i386-jos-elf-', set your TOOLPREFIX" 1>&2; \
	echo "*** environment variable to that prefix and run 'make' again." 1>&2; \
	echo "*** To turn off this error, run 'gmake TOOLPREFIX= ...'." 1>&2; \
	echo "***" 1>&2; exit 1; fi)
endif

# If the makefile can't find QEMU, specify its path here
# QEMU = qemu-system-i386

# Try to infer the correct QEMU
ifndef QEMU
QEMU = $(shell if which qemu > /dev/null; \
	then echo qemu; exit; \
	elif which qemu-system-i386 > /dev/null; \
	then echo qemu-system-i386; exit; \
	elif which qemu-system-x86_64 > /dev/null; \
```

```
	then echo qemu-system-x86_64; exit; \
	else \
	qemu=/Applications/Q.app/Contents/MacOS/i386-softmmu.app/Contents/MacOS/i386-softmmu; \
	if test -x $$qemu; then echo $$qemu; exit; fi; fi; \
	echo "***" 1>&2; \
	echo "*** Error: Couldn't find a working QEMU executable." 1>&2; \
	echo "*** Is the directory containing the qemu binary in your PATH" 1>&2; \
	echo "*** or have you tried setting the QEMU variable in Makefile?" 1>&2; \
	echo "***" 1>&2; exit 1)
endif

CC = $(TOOLPREFIX)gcc
AS = $(TOOLPREFIX)gas
LD = $(TOOLPREFIX)ld
OBJCOPY = $(TOOLPREFIX)objcopy
OBJDUMP = $(TOOLPREFIX)objdump
CFLAGS = -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer
#CFLAGS = -fno-pic -static -fno-builtin -fno-strict-aliasing -fvar-tracking -fvar-tracking-assignments -O0 -g -Wall -MD -gdwarf-2 -m32 -Werror -fno-omit-frame-pointer
CFLAGS += $(shell $(CC) -fno-stack-protector -E -x c /dev/null >/dev/null 2>&1 && echo -fno-stack-protector)
ASFLAGS = -m32 -gdwarf-2 -Wa,-divide
# FreeBSD ld wants ``elf_i386_fbsd"
LDFLAGS += -m $(shell $(LD) -V | grep elf_i386 2>/dev/null | head -n 1)

xv6.img: bootblock kernel fs.img
	dd if=/dev/zero of=xv6.img count=10000
	dd if=bootblock of=xv6.img conv=notrunc
```

```
	dd if=kernel of=xv6.img seek=1 conv=notrunc

xv6memfs.img: bootblock kernelmemfs
	dd if=/dev/zero of=xv6memfs.img count=10000
	dd if=bootblock of=xv6memfs.img conv=notrunc
	dd if=kernelmemfs of=xv6memfs.img seek=1 conv=notrunc

bootblock: bootasm.S bootmain.c
	$(CC) $(CFLAGS) -fno-pic -O -nostdinc -I. -c bootmain.c
	$(CC) $(CFLAGS) -fno-pic -nostdinc -I. -c bootasm.S
	$(LD) $(LDFLAGS) -N -e start -Ttext 0x7C00 -o bootblock.o bootasm.o bootmain.o
	$(OBJDUMP) -S bootblock.o > bootblock.asm
	$(OBJCOPY) -S -O binary -j .text bootblock.o bootblock
	./sign.pl bootblock

entryother: entryother.S
	$(CC) $(CFLAGS) -fno-pic -nostdinc -I. -c entryother.S
	$(LD) $(LDFLAGS) -N -e start -Ttext 0x7000 -o bootblockother.o entryother.o
	$(OBJCOPY) -S -O binary -j .text bootblockother.o entryother
	$(OBJDUMP) -S bootblockother.o > entryother.asm

initcode: initcode.S
	$(CC) $(CFLAGS) -nostdinc -I. -c initcode.S
	$(LD) $(LDFLAGS) -N -e start -Ttext 0 -o initcode.out initcode.o
	$(OBJCOPY) -S -O binary initcode.out initcode
	$(OBJDUMP) -S initcode.o > initcode.asm

kernel: $(OBJS) entry.o entryother initcode kernel.ld
```

```
	$(LD) $(LDFLAGS) -T kernel.ld -o kernel entry.o $(OBJS) -b binary initcode entryother
	$(OBJDUMP) -S kernel > kernel.asm
	$(OBJDUMP) -t kernel | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$$/d' > kernel.sym

# kernelmemfs is a copy of kernel that maintains the
# disk image in memory instead of writing to a disk.
# This is not so useful for testing persistent storage or
# exploring disk buffering implementations, but it is
# great for testing the kernel on real hardware without
# needing a scratch disk.
MEMFSOBJS = $(filter-out ide.o,$(OBJS)) memide.o
kernelmemfs: $(MEMFSOBJS) entry.o entryother initcode kernel.ld fs.img
	$(LD) $(LDFLAGS) -T kernel.ld -o kernelmemfs entry.o  $(MEMFSOBJS) -b binary initcode entryother fs.img
	$(OBJDUMP) -S kernelmemfs > kernelmemfs.asm
	$(OBJDUMP) -t kernelmemfs | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$$/d' > kernelmemfs.sym

tags: $(OBJS) entryother.S _init
	etags *.S *.c

vectors.S: vectors.pl
	perl vectors.pl > vectors.S

ULIB = ulib.o usys.o printf.o umalloc.o

_%: %.o $(ULIB)
	$(LD) $(LDFLAGS) -N -e main -Ttext 0 -o $@ $^
	$(OBJDUMP) -S $@ > $*.asm
```

```
	$(OBJDUMP) -t $@ | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$$/d' > $*.sym

_forktest: forktest.o $(ULIB)
	# forktest has less library code linked in - needs to be small
	# in order to be able to max out the proc table.
	$(LD) $(LDFLAGS) -N -e main -Ttext 0 -o _forktest forktest.o ulib.o usys.o
	$(OBJDUMP) -S _forktest > forktest.asm

mkfs: mkfs.c fs.h
	gcc -Werror -Wall -o mkfs mkfs.c

# Prevent deletion of intermediate files, e.g. cat.o, after first build, so
# that disk image changes after first build are persistent until clean.  More
# details:
# http://www.gnu.org/software/make/manual/html_node/Chained-Rules.html
.PRECIOUS: %.o

UPROGS=\
	_cat\
	_echo\
	_forktest\
	_grep\
	_init\
	_kill\
	_ln\
	_ls\
	_mkdir\
	_rm\
```

```
	_sh\
	_stressfs\
	_usertests\
	_wc\
  _cp\
	_zombie\

fs.img: mkfs README $(UPROGS)
	./mkfs fs.img README $(UPROGS)

-include *.d

clean:
	rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
	*.o *.d *.asm *.sym vectors.S bootblock entryother \
	initcode initcode.out kernel xv6.img fs.img kernelmemfs mkfs \
	.gdbinit \
	$(UPROGS)

# make a printout
FILES = $(shell grep -v '^\#' runoff.list)
PRINT = runoff.list runoff.spec README toc.hdr toc.ftr $(FILES)

xv6.pdf: $(PRINT)
	./runoff
	ls -l xv6.pdf

print: xv6.pdf
```

```
# run in emulators

bochs : fs.img xv6.img
	if [ ! -e .bochsrc ]; then ln -s dot-bochsrc .bochsrc; fi
	bochs -q

# try to generate a unique GDB port
GDBPORT = $(shell expr `id -u` % 5000 + 25000)
# QEMU's gdb stub command line changed in 0.11
QEMUGDB = $(shell if $(QEMU) -help | grep -q '^-gdb'; \
	then echo "-gdb tcp::$(GDBPORT)"; \
	else echo "-s -p $(GDBPORT)"; fi)
ifndef CPUS
CPUS := 2
endif
QEMUOPTS = -drive file=fs.img,index=1,media=disk,format=raw -drive
file=xv6.img,index=0,media=disk,format=raw -smp $(CPUS) -m 512 $(QEMUEXTRA)

qemu: fs.img xv6.img
	$(QEMU) -serial mon:stdio $(QEMUOPTS)

qemu-memfs: xv6memfs.img
	$(QEMU) -drive file=xv6memfs.img,index=0,media=disk,format=raw -smp $(CPUS) -m
256

qemu-nox: fs.img xv6.img
	$(QEMU) -nographic $(QEMUOPTS)
```

```
.gdbinit: .gdbinit.tmpl
	sed "s/localhost:1234/localhost:$(GDBPORT)/" < $^ > $@

qemu-gdb: fs.img xv6.img .gdbinit
	@echo "*** Now run 'gdb'." 1>&2
	$(QEMU) -serial mon:stdio $(QEMUOPTS) -S $(QEMUGDB)

qemu-nox-gdb: fs.img xv6.img .gdbinit
	@echo "*** Now run 'gdb'." 1>&2
	$(QEMU) -nographic $(QEMUOPTS) -S $(QEMUGDB)

# CUT HERE
# prepare dist for students
# after running make dist, probably want to
# rename it to rev0 or rev1 or so on and then
# check in that version.

EXTRA=\
	mkfs.c ulib.c user.h cat.c echo.c forktest.c grep.c kill.c\
	ln.c ls.c mkdir.c rm.c stressfs.c usertests.c wc.c zombie.c\
	printf.c umalloc.c cp.c\
	README dot-bochsrc *.pl toc.* runoff runoff1 runoff.list\
	.gdbinit.tmpl gdbutil\

dist:
	rm -rf dist
	mkdir dist
	for i in $(FILES); \
```

```
	do \
		grep -v PAGEBREAK $$i >dist/$$i; \
	done
	sed '/CUT HERE/,$$d' Makefile >dist/Makefile
	echo >dist/runoff.spec
	cp $(EXTRA) dist

dist-test:
	rm -rf dist
	make dist
	rm -rf dist-test
	mkdir dist-test
	cp dist/* dist-test
	cd dist-test; $(MAKE) print
	cd dist-test; $(MAKE) bochs || true
	cd dist-test; $(MAKE) qemu

# update this rule (change rev#) when it is time to
# make a new revision.
tar:
	rm -rf /tmp/xv6
	mkdir -p /tmp/xv6
	cp dist/* dist/.gdbinit.tmpl /tmp/xv6
	(cd /tmp; tar cf - xv6) | gzip >xv6-rev10.tar.gz  # the next one will be 10 (9/17)

.PHONY: dist-test dist

cp.c:
```

```
#include "types.h"
#include "stat.h"
#include "user.h"
#include "fcntl.h"

char buf[512];

int
main(int argc, char *argv[])
{
  int fd0, fd1, n;

  if(argc < 3){
    printf(1, "Need at least 2 arguments!\n");
    exit();
  }



  for (int i = 2; i < argc; i++){

    if((fd0 = open(argv[1], O_RDONLY)) < 0){
      printf(1, "cp: cannot open %s\n", argv[1]);
      exit();
    }

    if((fd1 = open(argv[i], O_CREATE|O_RDWR)) < 0){
      printf(1, "cp: cannot open %s\n", argv[i]);
```

```
    exit();
  }

  while ( (n = read (fd0, buf, sizeof(buf))) > 0 ){
    write(fd1, buf, n);
  }

  close(fd1);
  close(fd0);
 }

 exit();
}
```

I believe the makefile and cp.c were all that was needed to implement the cp command.

Script of the 2 variable cp command (I accidentally copied the wrong script file last time, the correct file was cp_command.txt rather than lab_1_cp_command.txt, which was what I copied before):

Script started on 2020-04-09 17:39:48-07:00 [TERM="xterm" TTY="/dev/pts/4" COLUMNS="140" LINES="35"]

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6

[006198682@csusb.edu@jb358-7 xv6]$ make qemu-nox

which: no qemu in (/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-

eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-

linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

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

c [?7l [2J [0mSeaBIOS (version 1.12.0-2.fc30)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF91280+1FED1280 C980

Press Ctrl-B to configure iPXE (PCI 00:03.0)...

Booting from Hard Disk...

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

$ ls

```
.              1 1 512
..             1 1 512
README         2 2 2290
cat            2 3 16132
echo           2 4 14972
forktest       2 5 9296
grep           2 6 18328
init           2 7 15576
kill           2 8 15000
ln             2 9 14872
ls             2 10 17496
mkdir          2 11 15116
rm             2 12 15092
sh             2 13 27616
stressfs       2 14 16008
usertests      2 15 66944
wc             2 16 16856
cp             2 17 15820
zombie         2 18 14688
console        3 19 0
$ cp README myFile1 myFile2
$ ls
.              1 1 512
..             1 1 512
README         2 2 2290
cat            2 3 16132
echo           2 4 14972
forktest       2 5 9296
```

grep 2 6 18328

init 2 7 15576

kill 2 8 15000

ln 2 9 14872

ls 2 10 17496

mkdir 2 11 15116

rm 2 12 15092

sh 2 13 27616

stressfs 2 14 16008

usertests 2 15 66944

wc 2 16 16856

cp 2 17 15820

zombie 2 18 14688

console 3 19 0

myFile1 2 20 2290

myFile2 2 21 2290

$ cat myFile1

xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix Version 6 (v6). xv6 loosely follows the structure and style of v6, but is implemented for a modern x86-based multiprocessor using ANSI C.

ACKNOWLEDGMENTS

xv6 is inspired by John Lions's Commentary on UNIX 6th Edition (Peer to Peer Communications; ISBN: 1-57398-013-7; 1st edition (June 14, 2000)). See also http://pdos.csail.mit.edu/6.828/2016/xv6.html, which provides pointers to on-line resources for v6.

xv6 borrows code from the following sources:

JOS (asm.h, elf.h, mmu.h, bootasm.S, ide.c, console.c, and others)

Plan 9 (entryother.S, mp.h, mp.c, lapic.c)

FreeBSD (ioapic.c)

NetBSD (console.c)

The following people have made contributions: Russ Cox (context switching, locking), Cliff Frey (MP), Xiao Yu (MP), Nickolai Zeldovich, and Austin Clements.

We are also grateful for the bug reports and patches contributed by Silas Boyd-Wickizer, Anton Burtsev, Cody Cutler, Mike CAT, Tej Chajed, Nelson Elhage, Saar Ettinger, Alice Ferrazzi, Nathaniel Filardo, Peter Froehlich, Yakir Goaron, Shivam Handa, Bryan Henry, Jim Huang, Alexander Kapshuk, Anders Kaseorg, kehao95, Wolfgang Keller, Eddie Kohler, Austin Liew, Imbar Marinescu, Yandong Mao, Hitoshi Mitake, Carmi Merimovich, Joel Nider, Greg Price, Ayan Shafqat, Eldar Sehayek, Yongming Shen, Cam Tenny, Rafael Ubal, Warren Toomey, Stephen Tu, Pablo Ventura, Xi Wang, Keiichi Watanabe, Nicolas Wolovick, Grant Wu, Jindong Zhang, Icenowy Zheng, and Zou Chang Wei.

ERROR REPORTS

Please send errors and suggestions to Frans Kaashoek and Robert Morris (kaashoek,rtm@mit.edu). The main purpose of xv6 is as a teaching operating system for MIT's 6.828, so we are more interested in

simplifications and clarifications than new features.

BUILDING AND RUNNING XV6

To build xv6 on an x86 ELF machine (like Linux or FreeBSD), run
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries. See http://pdos.csail.mit.edu/6.828/2016/tools.html.
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
simulator and run "make qemu".
$ cat mFile2
cat: cannot open mFile2
$ cat myFile2
xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix
Version 6 (v6). xv6 loosely follows the structure and style of v6,
but is implemented for a modern x86-based multiprocessor using ANSI C.

ACKNOWLEDGMENTS

xv6 is inspired by John Lions's Commentary on UNIX 6th Edition (Peer
to Peer Communications; ISBN: 1-57398-013-7; 1st edition (June 14,
2000)). See also http://pdos.csail.mit.edu/6.828/2016/xv6.html, which
provides pointers to on-line resources for v6.

xv6 borrows code from the following sources:
JOS (asm.h, elf.h, mmu.h, bootasm.S, ide.c, console.c, and others)
Plan 9 (entryother.S, mp.h, mp.c, lapic.c)
FreeBSD (ioapic.c)

NetBSD (console.c)

The following people have made contributions: Russ Cox (context switching, locking), Cliff Frey (MP), Xiao Yu (MP), Nickolai Zeldovich, and Austin Clements.

We are also grateful for the bug reports and patches contributed by Silas Boyd-Wickizer, Anton Burtsev, Cody Cutler, Mike CAT, Tej Chajed, Nelson Elhage, Saar Ettinger, Alice Ferrazzi, Nathaniel Filardo, Peter Froehlich, Yakir Goaron, Shivam Handa, Bryan Henry, Jim Huang, Alexander Kapshuk, Anders Kaseorg, kehao95, Wolfgang Keller, Eddie Kohler, Austin Liew, Imbar Marinescu, Yandong Mao, Hitoshi Mitake, Carmi Merimovich, Joel Nider, Greg Price, Ayan Shafqat, Eldar Sehayek, Yongming Shen, Cam Tenny, Rafael Ubal, Warren Toomey, Stephen Tu, Pablo Ventura, Xi Wang, Keiichi Watanabe, Nicolas Wolovick, Grant Wu, Jindong Zhang, Icenowy Zheng, and Zou Chang Wei.

ERROR REPORTS

Please send errors and suggestions to Frans Kaashoek and Robert Morris (kaashoek,rtm@mit.edu). The main purpose of xv6 is as a teaching operating system for MIT's 6.828, so we are more interested in simplifications and clarifications than new features.

BUILDING AND RUNNING XV6

To build xv6 on an x86 ELF machine (like Linux or FreeBSD), run
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries. See http://pdos.csail.mit.edu/6.828/2016/tools.html.
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
simulator and run "make qemu".
$ QEMU: Terminated

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6
[006198682@csusb.edu@jb358-7 xv6]$ exit

Script done on 2020-04-09 17:40:25-07:00 [COMMAND_EXIT_CODE="0"]

/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D

S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

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

c [?7l [2J [0mSeaBIOS (version 1.12.0-2.fc30)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF91280+1FED1280 C980

Press Ctrl-B to configure iPXE (PCI 00:03.0)...

Booting from Hard Disk..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

$ cp

Need at least 2 arguments!

$ cp

Need at least 2 arguments!

$ ls

. 1 1 512

.. 1 1 512

```
README          2 2 2290
cat             2 3 16132
echo            2 4 14972
forktest        2 5 9296
grep            2 6 18328
init            2 7 15576
kill            2 8 15000
ln              2 9 14872
ls              2 10 17496
mkdir           2 11 15116
rm              2 12 15092
sh              2 13 27616
stressfs        2 14 16008
usertests       2 15 66944
wc              2 16 16856
cp              2 17 15768
zombie          2 18 14688
console         3 19 0
$ cp README test
$ ls
.               1 1 512
..              1 1 512
README          2 2 2290
cat             2 3 16132
echo            2 4 14972
forktest        2 5 9296
grep            2 6 18328
init            2 7 15576
```

```
kill            2 8 15000
ln              2 9 14872
ls              2 10 17496
mkdir           2 11 15116
rm              2 12 15092
sh              2 13 27616
stressfs        2 14 16008
usertests       2 15 66944
wc              2 16 16856
cp              2 17 15768
zombie          2 18 14688
console         3 19 0
test            2 20 2290
$ cp README test1 test2
$ ls
.               1 1 512
..              1 1 512
README          2 2 2290
cat             2 3 16132
echo            2 4 14972
forktest        2 5 9296
grep            2 6 18328
init            2 7 15576
kill            2 8 15000
ln              2 9 14872
ls              2 10 17496
mkdir           2 11 15116
rm              2 12 15092
```

```
sh              2 13 27616
stressfs        2 14 16008
usertests       2 15 66944
wc              2 16 16856
cp              2 17 15768
zombie          2 18 14688
console         3 19 0
test            2 20 2290
test1           2 21 2290
test2           2 22 0
$ rm test
$ rm test1
$ rm test2
$ QEMU: Terminated

   ]0;006198682@csusb.edu@jb358-7:~/cse461/xv6
[006198682@csusb.edu@jb358-7 xv6]$ cd ..
   ]0;006198682@csusb.edu@jb358-7:~/cse461
[006198682@csusb.edu@jb358-7 cse461]$ git pull
remote: Enumerating objects: 7, done.   [K
remote: Counting objects:  14% (1/7)   [K
remote: Counting objects:  28% (2/7)   [K
remote: Counting objects:  42% (3/7)   [K
remote: Counting objects:  57% (4/7)   [K
remote: Counting objects:  71% (5/7)   [K
remote: Counting objects:  85% (6/7)   [K
remote: Counting objects: 100% (7/7)   [K
remote: Counting objects: 100% (7/7), done.   [K
```

```
remote: Compressing objects: 100% (1/1)   [K
remote: Compressing objects: 100% (1/1), done.   [K
remote: Total 4 (delta 3), reused 4 (delta 3), pack-reused 0   [K
Unpacking objects:  25% (1/4)
Unpacking objects:  50% (2/4)
Unpacking objects:  75% (3/4)
Unpacking objects: 100% (4/4)
Unpacking objects: 100% (4/4), done.
From https://github.com/DehNutCase/CSE-461
   05b747e..b47ff51  master     -> origin/master
Updating 05b747e..b47ff51
Fast-forward
 xv6/cp.c | 13    [32m++++++++   [m   [31m-----    [m
 1 file changed, 8 insertions(+), 5 deletions(-)
   ]0;006198682@csusb.edu@jb358-7:~/cse461
[006198682@csusb.edu@jb358-7 cse461]$ cd xv6
   ]0;006198682@csusb.edu@jb358-7:~/cse461/xv6
[006198682@csusb.edu@jb358-7 xv6]$ make
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector    -c -o cp.o cp.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _cp cp.o ulib.o usys.o printf.o umalloc.o
objdump -S _cp > cp.asm
objdump -t _cp | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > cp.sym
./mkfs fs.img README _cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs _usertests _wc _cp _zombie
nmeta 59 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 1) blocks 941 total 1000
balloc: first 694 blocks have been allocated
balloc: write bitmap block at sector 58
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, 0.480126 s, 10.7 MB/s

dd if=bootblock of=xv6.img conv=notrunc

1+0 records in

1+0 records out

512 bytes copied, 0.00255163 s, 201 kB/s

dd if=kernel of=xv6.img seek=1 conv=notrunc

416+1 records in

416+1 records out

213104 bytes (213 kB, 208 KiB) copied, 0.00803445 s, 26.5 MB/s

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6

[006198682@csusb.edu@jb358-7 xv6]$ cd ..

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ ls

[0m [01;32maShell [0m cp_command.txt debugging_script_2.txt SimpleShell.c [01;34mxv6 [0m xv6_2.txt

aShell.txt debugging_script_1.txt lab1_cp_command.txt simple_shell.txt xv6_1.txt

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ script cp_command.txt

Script started, file is cp_command.txt

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6

[006198682@csusb.edu@jb358-7 xv6]$ make qemu-nox

which: no qemu in (/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-

eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-

linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

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

c [?7l [2J [0mSeaBIOS (version 1.12.0-2.fc30)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF91280+1FED1280 C980

Press Ctrl-B to configure iPXE (PCI 00:03.0)...

Booting from Hard Disk...

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

$ ls

```
.              1 1 512
..             1 1 512
README         2 2 2290
cat            2 3 16132
echo           2 4 14972
forktest       2 5 9296
grep           2 6 18328
init           2 7 15576
kill           2 8 15000
ln             2 9 14872
ls             2 10 17496
mkdir          2 11 15116
rm             2 12 15092
sh             2 13 27616
stressfs       2 14 16008
usertests      2 15 66944
wc             2 16 16856
cp             2 17 15820
zombie         2 18 14688
console        3 19 0
$ cp README myFile1 myFile2
$ ls
.              1 1 512
..             1 1 512
README         2 2 2290
cat            2 3 16132
echo           2 4 14972
forktest       2 5 9296
```

grep 2 6 18328

init 2 7 15576

kill 2 8 15000

ln 2 9 14872

ls 2 10 17496

mkdir 2 11 15116

rm 2 12 15092

sh 2 13 27616

stressfs 2 14 16008

usertests 2 15 66944

wc 2 16 16856

cp 2 17 15820

zombie 2 18 14688

console 3 19 0

myFile1 2 20 2290

myFile2 2 21 2290

$ cat myFile1

xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix Version 6 (v6). xv6 loosely follows the structure and style of v6, but is implemented for a modern x86-based multiprocessor using ANSI C.

ACKNOWLEDGMENTS

xv6 is inspired by John Lions's Commentary on UNIX 6th Edition (Peer to Peer Communications; ISBN: 1-57398-013-7; 1st edition (June 14, 2000)). See also http://pdos.csail.mit.edu/6.828/2016/xv6.html, which provides pointers to on-line resources for v6.

xv6 borrows code from the following sources:

JOS (asm.h, elf.h, mmu.h, bootasm.S, ide.c, console.c, and others)

Plan 9 (entryother.S, mp.h, mp.c, lapic.c)

FreeBSD (ioapic.c)

NetBSD (console.c)

The following people have made contributions: Russ Cox (context switching, locking), Cliff Frey (MP), Xiao Yu (MP), Nickolai Zeldovich, and Austin Clements.

We are also grateful for the bug reports and patches contributed by Silas Boyd-Wickizer, Anton Burtsev, Cody Cutler, Mike CAT, Tej Chajed, Nelson Elhage, Saar Ettinger, Alice Ferrazzi, Nathaniel Filardo, Peter Froehlich, Yakir Goaron, Shivam Handa, Bryan Henry, Jim Huang, Alexander Kapshuk, Anders Kaseorg, kehao95, Wolfgang Keller, Eddie Kohler, Austin Liew, Imbar Marinescu, Yandong Mao, Hitoshi Mitake, Carmi Merimovich, Joel Nider, Greg Price, Ayan Shafqat, Eldar Sehayek, Yongming Shen, Cam Tenny, Rafael Ubal, Warren Toomey, Stephen Tu, Pablo Ventura, Xi Wang, Keiichi Watanabe, Nicolas Wolovick, Grant Wu, Jindong Zhang, Icenowy Zheng, and Zou Chang Wei.

ERROR REPORTS

Please send errors and suggestions to Frans Kaashoek and Robert Morris (kaashoek,rtm@mit.edu). The main purpose of xv6 is as a teaching operating system for MIT's 6.828, so we are more interested in

simplifications and clarifications than new features.

BUILDING AND RUNNING XV6

To build xv6 on an x86 ELF machine (like Linux or FreeBSD), run
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries. See http://pdos.csail.mit.edu/6.828/2016/tools.html.
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
simulator and run "make qemu".
$ cat mFile2
cat: cannot open mFile2
$ cat myFile2
xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix
Version 6 (v6). xv6 loosely follows the structure and style of v6,
but is implemented for a modern x86-based multiprocessor using ANSI C.

ACKNOWLEDGMENTS

xv6 is inspired by John Lions's Commentary on UNIX 6th Edition (Peer
to Peer Communications; ISBN: 1-57398-013-7; 1st edition (June 14,
2000)). See also http://pdos.csail.mit.edu/6.828/2016/xv6.html, which
provides pointers to on-line resources for v6.

xv6 borrows code from the following sources:
JOS (asm.h, elf.h, mmu.h, bootasm.S, ide.c, console.c, and others)
Plan 9 (entryother.S, mp.h,

Original Lab Report below:

Name: Ken Lin

SID: 006198682

Github repository of my code : https://github.com/DehNutCase/CSE-461

Part 1:

Simple Shell log:

I don’t know why ls didn’t work for the first run but did for the second.

Script started on 2020-04-09 17:04:01-07:00 [TERM="xterm" TTY="/dev/pts/2" COLUMNS="80" LINES="24"]

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ ./aShell

[1;1H[2J#ls

ls: cannot access "$'\273\377\177': No such file or directory

#

ls: cannot access "$'\273\377\177': No such file or directory

#help

#

#ls

ls: cannot access "$'\273\377\177': No such file or directory

#exit

#exit

#exit

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ ./aShell

[1;1H[2J#ls

SimpleShell.c cp_command.txt lab1_cp_command.txt xv6_1.txt

```
aShell        debugging_script_1.txt  simple_shell.txt     xv6_2.txt
aShell.txt     debugging_script_2.txt  xv6
#ls
SimpleShell.c  cp_command.txt              lab1_cp_command.txt  xv6_1.txt
aShell        debugging_script_1.txt  simple_shell.txt     xv6_2.txt
aShell.txt     debugging_script_2.txt  xv6
#ls
SimpleShell.c  cp_command.txt              lab1_cp_command.txt  xv6_1.txt
aShell        debugging_script_1.txt  simple_shell.txt     xv6_2.txt
aShell.txt     debugging_script_2.txt  xv6
#
SimpleShell.c  cp_command.txt              lab1_cp_command.txt  xv6_1.txt
aShell        debugging_script_1.txt  simple_shell.txt     xv6_2.txt
aShell.txt     debugging_script_2.txt  xv6
#
SimpleShell.c  cp_command.txt              lab1_cp_command.txt  xv6_1.txt
aShell        debugging_script_1.txt  simple_shell.txt     xv6_2.txt
aShell.txt     debugging_script_2.txt  xv6
#exit
]0;006198682@csusb.edu@jb358-7:~/cse461
[006198682@csusb.edu@jb358-7 cse461]$ ./aShell
[1;1H[2J#
#ls
SimpleShell.c  cp_command.txt              lab1_cp_command.txt  xv6_1.txt
aShell        debugging_script_1.txt  simple_shell.txt     xv6_2.txt
aShell.txt     debugging_script_2.txt  xv6
#help
#ls
```

ls: cannot access 'p': No such file or directory

#

ls: cannot access 'p': No such file or directory

#exit

#ls

ls: cannot access 'p': No such file or directory

#exit

#exit

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ e[K

exit

Script done on 2020-04-09 17:04:58-07:00 [COMMAND_EXIT_CODE="0"]

Part 2:

Xv6 debugger mode, disassembling kernel.

Log of debugger:

Script started on 2020-04-09 15:04:11-07:00 [TERM="xterm" TTY="/dev/pts/4" COLUMNS="114" LINES="29"]

]0;006198682@csusb.edu@jb358-7:~/cse461[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ gdb

[35;1mGNU gdb (GDB) Fedora 8.3-6.fc30

[mCopyright (C) 2019 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-redhat-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".

warning: File "/home/csusb.edu/006198682/cse461/xv6/.gdbinit" auto-loading has been declined by your `auto-load safe-path' set to "$debugdir:$datadir/auto-load:/usr/lib/golang/src/runtime/runtime-gdb.py".

To enable execution of this file add

add-auto-load-safe-path /home/csusb.edu/006198682/cse461/xv6/.gdbinit

line to your configuration file "/home/csusb.edu/006198682/.gdbinit".

To completely disable this security protection add

set auto-load safe-path /

line to your configuration file "/home/csusb.edu/006198682/.gdbinit".

For more information about this security protection see the

"Auto-loading safe path" section in the GDB manual. E.g., run from the shell:

info "(gdb)Auto-loading safe path"

(gdb) target remote 27604

27604: No such file or directory.

(gdb) tr[Karget remote :27604

Remote debugging using :27604

warning: Remote gdbserver does not support determining executable automatically.

RHEL <=6.8 and <=7.2 versions of gdbserver do not support such automatic executable detection.

The following versions of gdbserver support it:

- Upstream version of gdbserver (unsupported) 7.10 or later

- Red Hat Developer Toolset (DTS) version of gdbserver from DTS 4.0 or later (only on x86_64)
- RHEL-7.3 versions of gdbserver (on any architecture)

warning: No executable has been specified and target does not support

determining executable automatically. Try using the "file" command.

[34m0x0000fff0[m in [33m??[m ()

(gdb) continue

Continuing.

file   ^C

Thread 1 received signal SIGINT, Interrupt.

[34m0x801037e1[m in [33m??[m ()

(gdb) file kernel

A program is being debugged already.

Are you sure you want to change the file? (y or n) y

Reading symbols from [32mkernel[m...

(gdb) set disassembly-flavor intel

(gdb) disass

Dump of assembler code for function mycpu:

   [34m0x801037d0[m <+0>: push   ebp

   [34m0x801037d1[m <+1>: mov   ebp,esp

   [34m0x801037d3[m <+3>: push   esi

   [34m0x801037d4[m <+4>: push   ebx

   [34m0x801037d5[m <+5>: pushf

   [34m0x801037d6[m <+6>: pop   eax

   [34m0x801037d7[m <+7>: test   ah,0x2

   [34m0x801037da[m <+10>:   jne   0x8010382a <mycpu+90>

   [34m0x801037dc[m <+12>:   call   0x80102800 <lapicid>

=> [34m0x801037e1[m <+17>:   mov   esi,DWORD PTR ds:0x80112d00

   [34m0x801037e7[m <+23>:   mov   ebx,eax

```
   [34m0x801037e9[m <+25>:     test   esi,esi
   [34m0x801037eb[m <+27>:     jle    0x8010381d <mycpu+77>
   [34m0x801037ed[m <+29>:     xor    edx,edx
   [34m0x801037ef[m <+31>:jmp    0x801037ff <mycpu+47>
   [34m0x801037f1[m <+33>:     lea    esi,[esi+eiz*1+0x0]
   [34m0x801037f8[m <+40>:     add    edx,0x1
   [34m0x801037fb[m <+43>:     cmp    edx,esi
   [34m0x801037fd[m <+45>:     je     0x8010381d <mycpu+77>
   [34m0x801037ff[m <+47>: imul   ecx,edx,0xb0
   [34m0x80103805[m <+53>:     movzx  eax,BYTE PTR [ecx-0x7feed880]
   [34m0x8010380c[m <+60>:     cmp    eax,ebx
   [34m0x8010380e[m <+62>:     jne    0x801037f8 <mycpu+40>
   [34m0x80103810[m <+64>:     lea    esp,[ebp-0x8]
   [34m0x80103813[m <+67>:     lea    eax,[ecx-0x7feed880]
   [34m0x80103819[m <+73>:     pop    ebx
   [34m0x8010381a[m <+74>:     pop    esi
[m--Type <RET> for more, q to quit, c to continue without paging--
   [34m0x8010381b[m <+75>:     pop    ebp
   [34m0x8010381c[m <+76>:     ret
   [34m0x8010381d[m <+77>:     sub    esp,0xc
   [34m0x80103820[m <+80>:     push   0x801073c7
   [34m0x80103825[m <+85>:     call   0x80100380 <panic>
   [34m0x8010382a[m <+90>:     sub    esp,0xc
   [34m0x8010382d[m <+93>:     push   0x801074a4
   [34m0x80103832[m <+98>:     call   0x80100380 <panic>
End of assembler dump.
(gdb) Quit
(gdb) quit
```

A debugging session is active.

Inferior 1 [Remote target] will be detached.

Quit anyway? (y or n) y

Detaching from program: /home/csusb.edu/006198682/cse461/xv6/kernel, Remote target

Ending remote debugging.

[Inferior 1 (Remote target) detached]

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ exit

Script done on 2020-04-09 15:05:47-07:00 [COMMAND_EXIT_CODE="0"]

Log of xv6:

Script started on 2020-04-09 15:04:07-07:00 [TERM="xterm" TTY="/dev/pts/0" COLUMNS="113" LINES="30"]

]0;006198682@csusb.edu@jb358-7:~/cse461[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ make qemu-nox-gdb

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, 0.645025 s, 7.9 MB/s

dd if=bootblock of=xv6.img conv=notrunc

1+0 records in

1+0 records out

512 bytes copied, 0.001909 s, 268 kB/s

dd if=kernel of=xv6.img seek=1 conv=notrunc

416+1 records in

416+1 records out

213104 bytes (213 kB, 208 KiB) copied, 0.0071189 s, 29.9 MB/s

which: no qemu in
(/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin
:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchai
n/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-

3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

which: no qemu in (/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/

Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

*** 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::27604

c[?7l[2J[0mSeaBIOS (version 1.12.0-2.fc30)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF91280+1FED1280 C980

Press Ctrl-B to configure iPXE (PCI 00:03.0)...

Booting from Hard Disk...

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

$ ex ls

.   1 1 512

..   1 1 512

README   2 2 2290

cat   2 3 16132

echo 2 4 14972

forktest 2 5 9296

grep 2 6 18328

init 2 7 15576

kill 2 8 15000

ln 2 9 14872

ls 2 10 17496

mkdir 2 11 15116

rm 2 12 15092

sh 2 13 27616

stressfs 2 14 16008

usertests 2 15 66944

wc 2 16 16856

zombie 2 17 14688

console 3 18 0

$ exit

exec: fail

exec exit failed

$ QEMU: Terminated

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ exit

Script done on 2020-04-09 15:05:45-07:00 [COMMAND_EXIT_CODE="0"]

Part 3:

Debugging commands and observations:

The debugger isn't very useful if I don't have the source code in front of me, since there's no way I'd know what swtch means or is supposed to do without the source code. Further, it's simpler for me to modify the code directly and change it directly and test the results than to use the debugger to do so.

While the debugger allows me access to the values of variables during run-time, I can obtain that information by simply making my program print to console, so it would only be useful if I ran into a bug I didn't expect or if I wasn't aware of where a bug was. However, in those cases running the debugger line by line would take, perhaps, just as long as looking over the source code line by line (possibly far longer, if the debugger is going through a loop in the code---a human can get away with reading the loop once, the debugger will have to keep going through the loop).

The debugger doesn't seem very useful for debugging, since it doesn't really do anything that directly modifying the source code and running test doesn't.

Debugger log:

Script started on 2020-04-09 14:41:54-07:00 [TERM="xterm" TTY="/dev/pts/4" COLUMNS="114" LINES="29"]

]0;006198682@csusb.edu@jb358-7:~/cse461[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ gdb

[35;1mGNU gdb (GDB) Fedora 8.3-6.fc30

[mCopyright (C) 2019 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-redhat-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".

warning: File "/home/csusb.edu/006198682/cse461/xv6/.gdbinit" auto-loading has been declined by your `auto-load safe-path' set to "$debugdir:$datadir/auto-load:/usr/lib/golang/src/runtime/runtime-gdb.py".

To enable execution of this file add

    add-auto-load-safe-path /home/csusb.edu/006198682/cse461/xv6/.gdbinit

line to your configuration file "/home/csusb.edu/006198682/.gdbinit".

To completely disable this security protection add

    set auto-load safe-path /

line to your configuration file "/home/csusb.edu/006198682/.gdbinit".

For more information about this security protection see the

"Auto-loading safe path" section in the GDB manual. E.g., run from the shell:

    info "(gdb)Auto-loading safe path"

(gdb) target remote :27604

Remote debugging using :27604

warning: Remote gdbserver does not support determining executable automatically.

RHEL <=6.8 and <=7.2 versions of gdbserver do not support such automatic executable detection.

The following versions of gdbserver support it:

- Upstream version of gdbserver (unsupported) 7.10 or later

- Red Hat Developer Toolset (DTS) version of gdbserver from DTS 4.0 or later (only on x86_64)

- RHEL-7.3 versions of gdbserver (on any architecture)

warning: No executable has been specified and target does not support

determining executable automatically. Try using the "file" command.

[34m0x0000fff0[m in [33m??[m ()

(gdb) file kernel

A program is being debugged already.

Are you sure you want to change the file? (y or n) y

Reading symbols from [32mkernel[m...

(gdb) break swtch

```
Breakpoint 1 at [34m0x801046eb[m: file [32mswtch.S[m, line 11.
(gdb) continue
Continuing.
[Switching to Thread 2]

Thread 2 hit Breakpoint 1, [33mswtch[m () at [32mswtch.S[m:11
11          movl [35m4[m[31m(%[m[32mesp[m[31m),[m [31m%[m[32meax[m
(gdb) step
12          movl [35m8[m[31m(%[m[32mesp[m[31m),[m [31m%[m[32medx[m
(gdb) step
15          pushl [31m%[m[32mebp[m
(gdb) step
[33mswtch[m () at [32mswtch.S[m:16
16          pushl [31m%[m[32mebx[m
(gdb) step
[33mswtch[m () at [32mswtch.S[m:17
17          pushl [31m%[m[32mesi[m
(gdb) step
[33mswtch[m () at [32mswtch.S[m:18
18          pushl [31m%[m[32medi[m
(gdb) step
[33mswtch[m () at [32mswtch.S[m:21
21          movl [31m%[m[32mesp[m[31m,[m [31m(%[m[32meax[m[31m)[m
(gdb) step
22          movl [31m%[m[32medx[m[31m,[m [31m%[m[32mesp[m
(gdb) step
[33mswtch[m () at [32mswtch.S[m:25
25          popl [31m%[m[32medi[m
```

(gdb) step

[33mswtch[m () at [32mswtch.S[m:26

26 popl [31m%[m[32mesi[m

(gdb) step

[33mswtch[m () at [32mswtch.S[m:27

27 popl [31m%[m[32mebx[m

(gdb) step

[33mswtch[m () at [32mswtch.S[m:28

28 popl [31m%[m[32mebp[m

(gdb) step

[33mswtch[m () at [32mswtch.S[m:29

29 [01;34mret[m

(gdb) step

[33mforkret[m () at [32mproc.c[m:401

401 [01mrelease[m[31m(&[mptable[31m.[mlock[31m);[m

(gdb) step

[33mrelease[m ([36mlk[m=0x80112d20 <ptable>) at [32mspinlock.c[m:49

49 [01;34mif[m[31m(![m[01mholding[m[31m([mlk[31m))[m

(gdb) step

[33mholding[m ([36mlock[m=0x80112d20 <ptable>) at [32mspinlock.c[m:92

92 [01;34mreturn[m lock[31m->[mlocked [31m&&[m lock[31m->[mcpu [31m==[m [01mmycpu[m[31m();[m

(gdb) clear

No breakpoint at this line.

(gdb) continue

Continuing.

Thread 2 hit Breakpoint 1, [33mswtch[m () at [32mswtch.S[m:11

11 movl [35m4[m[31m(%[m[32mesp[m[31m),[m [31m%[m[32meax[m

(gdb) clear

Deleted breakpoint 1

(gdb) continue

Continuing.

break exec

^C

Thread 2 received signal SIGINT, Interrupt.

[34m0x801037e1[m in [33mmycpu[m () at [32mproc.c[m:45

45 apicid [31m=[m [01mlapicid[m[31m();[m

(gdb) break exec

Breakpoint 2 at [34m0x80100a70[m: file [32mexec.c[m, line 20.

(gdb) continue

Continuing.

Thread 2 hit Breakpoint 2, [33mexec[m ([36mpath[m=0x18e0 "ls", [36margv[m=0x8dfbeed0) at [32mexec.c[m:20

20 [01;34mstruct[m [32mproc[m [31m*[mcurproc [31m=[m [01mmyproc[m[31m();[m

(gdb) conti[K[Ktinue

Continuing.

Thread 2 hit Breakpoint 2, [33mexec[m ([36mpath[m=0x18e0 "ls", [36margv[m=0x8df23ed0) at [32mexec.c[m:20

20 [01;34mstruct[m [32mproc[m [31m*[mcurproc [31m=[m [01mmyproc[m[31m();[m

(gdb) continue

Continuing.

[Switching to Thread 1]

Thread 1 hit Breakpoint 2, [33mexec[m ([36mpath[m=0x18e0 "ls", [36margv[m=0x8dfc6ed0) at [32mexec.c[m:20

20 [01;34mstruct[m [32mproc[m [31m*[mcurproc [31m=[m [01mmyproc[m[31m();[m

(gdb) continue

Continuing.

Remote connection closed

(gdb) exit

Undefined command: "exit". Try "help".

(gdb) quit

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ exit

Script done on 2020-04-09 14:46:35-07:00 [COMMAND_EXIT_CODE="0"]

Xv6 log:

Script started on 2020-04-09 14:42:05-07:00 [TERM="xterm" TTY="/dev/pts/3" COLUMNS="114" LINES="29"]

]0;006198682@csusb.edu@jb358-7:~/cse461[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ make qemu-nox-gdb

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, 0.355392 s, 14.4 MB/s

dd if=bootblock of=xv6.img conv=notrunc

1+0 records in

1+0 records out

512 bytes copied, 0.00182727 s, 280 kB/s

dd if=kernel of=xv6.img seek=1 conv=notrunc

416+1 records in

416+1 records out

213104 bytes (213 kB, 208 KiB) copied, 0.0045323 s, 47.0 MB/s

sed "s/localhost:1234/localhost:27604/" < .gdbinit.tmpl > .gdbinit

which: no qemu in
(/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin
:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchai
n/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-

linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

which: no qemu in (/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D

S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

*** 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::27604

c[?7l[2J[0mSeaBIOS (version 1.12.0-2.fc30)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF91280+1FED1280 C980

Press Ctrl-B to configure iPXE (PCI 00:03.0)...

Booting from Hard Disk..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

$ ls

. 1 1 512

.. 1 1 512

README 2 2 2290

cat 2 3 16132

echo 2 4 14972

```
forktest       2 5 9296
grep           2 6 18328
init           2 7 15576
kill           2 8 15000
ln             2 9 14872
ls             2 10 17496
mkdir          2 11 15116
rm             2 12 15092
sh             2 13 27616
stressfs       2 14 16008
usertests      2 15 66944
wc             2 16 16856
zombie         2 17 14688
console        3 18 0
$ ls
.              1 1 512
..             1 1 512
README         2 2 2290
cat            2 3 16132
echo           2 4 14972
forktest       2 5 9296
grep           2 6 18328
init           2 7 15576
kill           2 8 15000
ln             2 9 14872
ls             2 10 17496
mkdir          2 11 15116
rm             2 12 15092
```

sh 2 13 27616

stressfs 2 14 16008

usertests 2 15 66944

wc 2 16 16856

zombie 2 17 14688

console 3 18 0

$ ls -l

ls: cannot open -l

$ QEMU: Terminated

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6[006198682@csusb.edu@jb358-7 xv6]$ exit

Script done on 2020-04-09 14:46:37-07:00 [COMMAND_EXIT_CODE="0"]

Part 4:

CP command:

Log of XV6 running the cp command:

Script started on 2020-04-09 16:46:47-07:00 [TERM="xterm" TTY="/dev/pts/2" COLUMNS="116" LINES="27"]

]0;006198682@csusb.edu@jb358-7:~/cse461

[006198682@csusb.edu@jb358-7 cse461]$ cd xv6

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6

[006198682@csusb.edu@jb358-7 xv6]$ make qemu-0[Knox

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, 0.434345 s, 11.8 MB/s

dd if=bootblock of=xv6.img conv=notrunc

1+0 records in

1+0 records out

512 bytes copied, 0.00136823 s, 374 kB/s

dd if=kernel of=xv6.img seek=1 conv=notrunc

416+1 records in

416+1 records out

213104 bytes (213 kB, 208 KiB) copied, 0.00649258 s, 32.8 MB/s

which: no qemu in
(/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/lin/bin
:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/microblaze/linux_toolchai
n/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-
gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-
none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-
eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-
3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/
Chimera64-
1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_D
S/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/
Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-
linux/tools:/opt/android-sdk-linux/platform-
tools:/usr/java/latest/bin:/opt/anaconda3/bin:/opt/Xilinx/SDK/2018.2/bin:/opt/Xilinx/SDK/2018.
2/gnu/microblaze/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/arm/lin/bin:/opt/Xilinx/SDK/2018.2/gnu/
microblaze/linux_toolchain/lin64_le/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-linux-

gnueabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch32/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-linux/bin:/opt/Xilinx/SDK/2018.2/gnu/aarch64/lin/aarch64-none/bin:/opt/Xilinx/SDK/2018.2/gnu/armr5/lin/gcc-arm-none-eabi/bin:/opt/Xilinx/SDK/2018.2/tps/lnx64/cmake-3.3.2/bin:/opt/Xilinx/DocNav:/opt/Xilinx/Vivado/2018.2/bin:/usr/lib64/openmpi/bin:/opt/UCSF/Chimera64-1.12/bin:/usr/local/MATLAB/R2018a/bin:/share/bin:/usr/local/racket/bin:/opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE_DS/common/bin/lin64:/opt/Xilinx/Vivado/2017.2/bin:/opt/Xilinx/Vivado_HLS/2017.2/bin:/opt/android-studio/bin:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/java/latest/bin:/usr/lib64/qt-3.3/bin:/usr/share/Modules/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/csusb.edu/006198682/.local/bin:/home/csusb.edu/006198682/bin)

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

c[?7l[2J[0mSeaBIOS (version 1.12.0-2.fc30)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF91280+1FED1280 C980

Press Ctrl-B to configure iPXE (PCI 00:03.0)...

Booting from Hard Disk...

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

$ cp

Need 2 arguments!

$ ls

. 1 1 512

.. 1 1 512

README 2 2 2290

cat 2 3 16132

echo 2 4 14972

forktest 2 5 9296

grep 2 6 18328

init 2 7 15576

kill 2 8 15000

ln 2 9 14872

ls 2 10 17496

mkdir 2 11 15116

rm 2 12 15092

sh 2 13 27616

stressfs 2 14 16008

usertests 2 15 66944

wc 2 16 16856

cp 2 17 15444

zombie 2 18 14688

console 3 19 0

$ cp README test

$ ls

. 1 1 512

.. 1 1 512

README 2 2 2290

```
cat             2 3 16132
echo            2 4 14972
forktest        2 5 9296
grep            2 6 18328
init            2 7 15576
kill            2 8 15000
ln              2 9 14872
ls              2 10 17496
mkdir           2 11 15116
rm              2 12 15092
sh              2 13 27616
stressfs        2 14 16008
usertests       2 15 66944
wc              2 16 16856
cp              2 17 15444
zombie          2 18 14688
console         3 19 0
test            2 20 2290
$ cat test
xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix
Version 6 (v6).  xv6 loosely follows the structure and style of v6,
but is implemented for a modern x86-based multiprocessor using ANSI C.
```

ACKNOWLEDGMENTS

xv6 is inspired by John Lions's Commentary on UNIX 6th Edition (Peer to Peer Communications; ISBN: 1-57398-013-7; 1st edition (June 14, 2000)). See also http://pdos.csail.mit.edu/6.828/2016/xv6.html, which

provides pointers to on-line resources for v6.

xv6 borrows code from the following sources:

JOS (asm.h, elf.h, mmu.h, bootasm.S, ide.c, console.c, and others)

Plan 9 (entryother.S, mp.h, mp.c, lapic.c)

FreeBSD (ioapic.c)

NetBSD (console.c)

The following people have made contributions: Russ Cox (context switching, locking), Cliff Frey (MP), Xiao Yu (MP), Nickolai Zeldovich, and Austin Clements.

We are also grateful for the bug reports and patches contributed by Silas Boyd-Wickizer, Anton Burtsev, Cody Cutler, Mike CAT, Tej Chajed, Nelson Elhage, Saar Ettinger, Alice Ferrazzi, Nathaniel Filardo, Peter Froehlich, Yakir Goaron, Shivam Handa, Bryan Henry, Jim Huang, Alexander Kapshuk, Anders Kaseorg, kehao95, Wolfgang Keller, Eddie Kohler, Austin Liew, Imbar Marinescu, Yandong Mao, Hitoshi Mitake, Carmi Merimovich, Joel Nider, Greg Price, Ayan Shafqat, Eldar Sehayek, Yongming Shen, Cam Tenny, Rafael Ubal, Warren Toomey, Stephen Tu, Pablo Ventura, Xi Wang, Keiichi Watanabe, Nicolas Wolovick, Grant Wu, Jindong Zhang, Icenowy Zheng, and Zou Chang Wei.

ERROR REPORTS

Please send errors and suggestions to Frans Kaashoek and Robert Morris

(kaashoek,rtm@mit.edu). The main purpose of xv6 is as a teaching
operating system for MIT's 6.828, so we are more interested in
simplifications and clarifications than new features.

BUILDING AND RUNNING XV6

To build xv6 on an x86 ELF machine (like Linux or FreeBSD), run
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries. See http://pdos.csail.mit.edu/6.828/2016/tools.html.
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
simulator and run "make qemu".
$ QEMU: Terminated

]0;006198682@csusb.edu@jb358-7:~/cse461/xv6
[006198682@csusb.edu@jb358-7 xv6]$ exit

Script done on 2020-04-09 16:47:12-07:00 [COMMAND_EXIT_CODE="0"]