**Homework 1**

Part 1: Hello World (20 points)

Write a program for xv6 that, when run, prints ”Hello world” to the xv6 console. This can be broken up

into a few steps:

1. Create a file in the xv6 directory named hello.c

2. Put code you need to implement printing ”Hello world” into hello.c

3. Edit the file Makefile, find the section UPROGS (which contains a list of programs to be built), and add

a line to tell it to build your Hello World program. When you’re done that portion of the Makefile

should look like:

UPROGS=\

\_cat\

\_echo\

\_forktest\

\_grep\

\_init\

\_kill\

\_ln\

\_ls\

\_mkdir\

\_rm\

\_sh\

\_stressfs\

\_usertests\

\_wc\

\_zombie\

\_hello\

4. Run make to build xv6, including your new program (repeating steps 2 and 4 until you have compiling

code)

5. Run make qemu to launch xv6, and then type hello in the QEMU window. You should see ”Hello

world” be printed out

Of course step 2 is where the bulk of the work lies. You will find that many things are subtly di↵erent

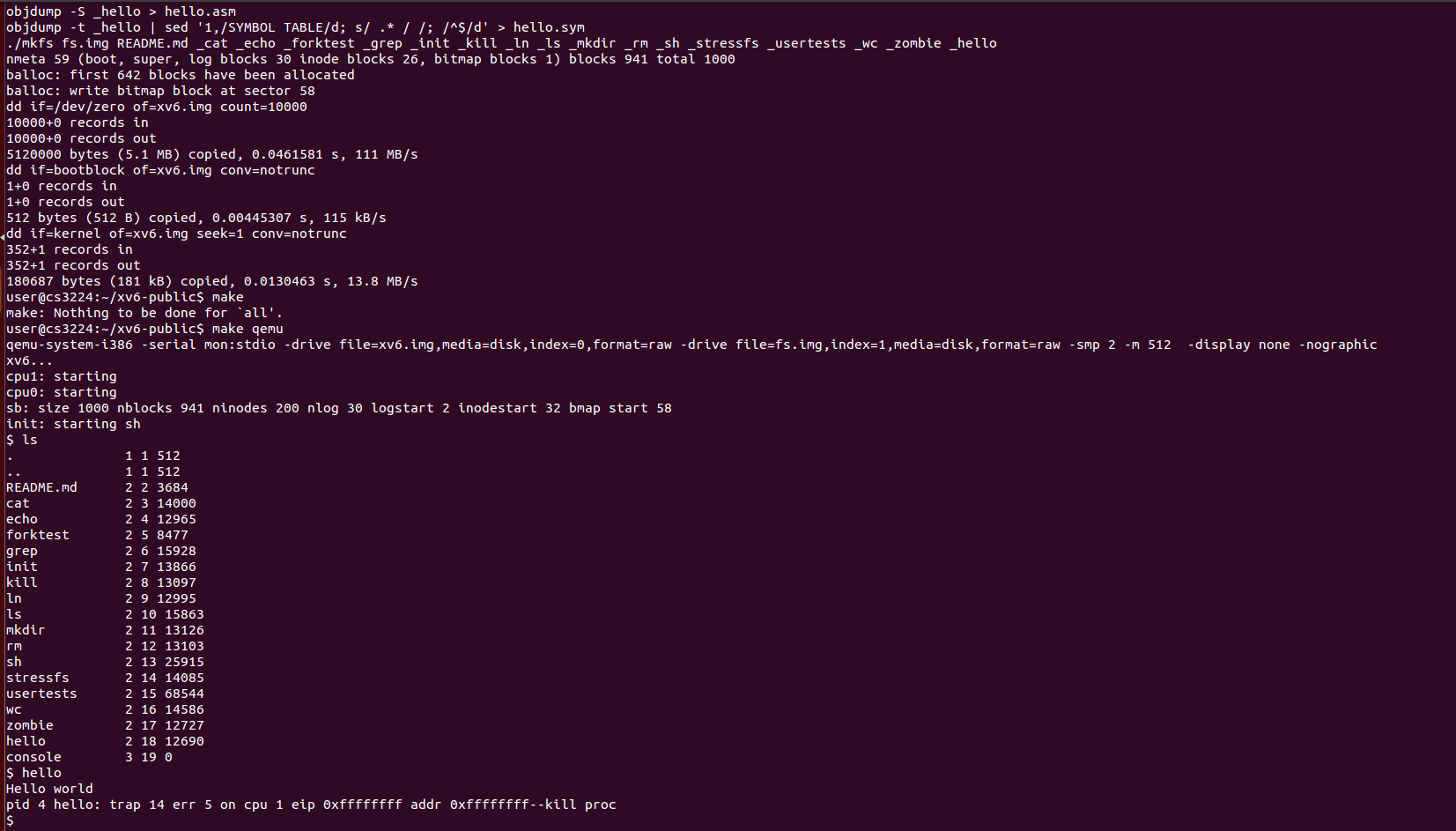
from the programming environments you’ve used before; for example, the printf function takes an extra

argument that specifies where it should print to. This is because you’re writing programs for a new operating

system, and it doesn’t have to follow the conventions of anything you’ve used before. To get a feel for how

programs look in xv6, and how various APIs should be called, you can look at the source code for other

utilities: echo.c, cat.c, wc.c, ls.c.



Part 2: Implementing the uniq command (50 points)

uniq is a Unix utility which, when fed a text file, outputs the file with adjacent identical lines collapsed to

one. If a filename is provided on the command line (i.e., uniq FILE) then uniq should open it, read, filter

out, print without repeated lines in this file, and then close it. If no filename is provided, uniq should read

from standard input.

Here’s an example of the basic usage of uniq: