```
Michael Kepple
Operating Systems
Lab 11

#2 - Write a "one line pipeline"
- "ls -l /dev | awk ’{print $5}’ | tr -d ’,’ | sort -n | uniq"

#4 - /dev/zero
- What happens when you output to /dev/zero? It acts as a sink, succeeding with
  no other effects. Essentially used to re-route output away from STDOUT.
- What happens when you read from it? Will return an infite sequence of zero
  bits, unless a size if specified (using dd for instance).
Modify:
What I did:
To make it output all ’1’ bits:
1) edited memory.c in /usr/src/drivers/memory
Diff output:
349c349
<           if ((s = sys_safememset(endpt, grant, 0, ’\0’, count)) != OK)
---
>           if ((s = sys_safememset(endpt, grant, 0, -1, count)) != OK)
2) make install
3) service refresh memory

To make it output ’0101010..."
1) edited memory.c in /usr/src/drivers/memory
Diff output:
349c349
<           if ((s = sys_safememset(endpt,grant, 0, ’\0’, count)) != OK)
---
>           if ((s = sys_safememset(endpt, grant, 0, 1431655765, count)) != OK)
2) make install
3) service refresh memory

To make it output all zeroes again:
1) restored my old copy of memory.c
2) make install
3) service refresh memory

# 5 - The "Hello, World" Device Driver
What I did:
Disply greeting in UPPERCASE:
1) edited hello.h in /usr/src/drivers/hello
Diff output:
5c5
< #define HELLO_MESSAGE "Hello, World!\n"
---
> #define HELLO_MESSAGE "HELLO, WORLD!\n"
2) make install
3) service refresh hello

Remove debug output:
1) edited hello.c in /usr/src/drivers/hello
Diff output:
47d46
<       printf("hello_open(). Called %d time(s).\n", ++open_counter);
53d51
<       printf("hello_close()\n");
70d67
<       printf("hello_transfer()\n");
2) make install
3) service down hello
4) service up /usr/sbin/hello -dev /dev/hello
```

```
    #6 - Magic Eight Ball Device Driver
    - Minix won’t let me SSH to student right now. As soon as I figure out the
      connection issues I can print out copies of my code.

    #7 - Writing to the "Hello, World!" device driver
    - See above
```