2.1.1

HINT 1 (1)

Remember the 1st chapter exercises when we were using the *yes* command and we observed the cache strategy of the OS. Consider how big the memory of your machine is (f.e. with *top*) !

2.1.2

HINT 1 (1)

Just compare how many write statements your program has to execute for 1b or 4k blocksize and conclude!

2.1.3

HINT 1 (1)

During which of the 10 measurements is actually your harddisk involved? You can look at a *vmstat* or *iostat* command in a separate window to verify it.

2.1.4

HINT 1 (2)

'dd' for large (empty) file creation

If you use 'dd' be careful! Modern file systems support sparse files. A 10 GB file created with:

```
bash> dd if=/dev/zero of=/tmp/10GB bs=1000
count=0 seek=$[1000*1000*10]
```

does not occupy blocks in the file system. You can verify this using the *stat* command.

If you use dd to create a physical existing file, you can do:

```
bash> dd if=/dev/zero of=/tmp/10GB bs=1000
count=$[1000*10]
```

2.1.4

HINT 2 (2)

Make sure that for the first measurement the buffer cache does not contain files you are going to read .

For the second measurement we have to be sure that the cache contains the files we are going to read ...

cat is a good 4k blocksize reader program if you redirect the output to ***/dev/null***

2.1.5

HINT 1 (1)

The time to read a single byte if disk heads are positioned is neglegible compared to the time to position the disk heads.

Therefore the seek time can be estimated as the average time to read 1 byte in this measurement!