q1.1

59 system calls

I asked chatgpt for examples of how to use each command and eventually used the following commands used to find different syscalls:

```
awk '{print $3}' apache2.strace | cut -d'(' -f1 | sort | uniq | wc -l
```

this output 61, but I printed out all the commands to get a glance and noticed the command caught --- and +++ so I subtracted 2. (also can filter it out with grep)

```
awk '{print $3}' apache2.strace | cut -d'(' -f1 | sort | uniq
```

q1.2:

Filtered out for only alphanumeric characters now
Command:

```
cat apache2.strace | awk '{print $3}' | cut -d'(' -f1 | \
sort | grep -E '^[a-zA-Z0-9_]+$' | uniq | wc -l
```

q1.3:

> 982 mmap
> 1001 newfstatat
> 1153 epoll_ctl
> 1447 read
> 4223 getrandom

q1.4

```
cat apache2.strace | awk '{print $3}' | cut -d'(' -f1 | sort | uniq -c | sort | tail -n 5
```

Output:

> 982 mmap
> 1001 newfstatat
> 1153 epoll_ctl
> 1447 read
> 4223 getrandom

q2.1

28 process ids

Using:

```
awk '{print $1}' apache2.strace | sort | uniq | wc -l
```

q2.2

The clone() system call can create new processes and return the PID (process ids)

q2.3

It is invoked 27 times shown by:
`awk '{print $3}' apache2.strace | cut -d'(' -f1 | grep clone | wc -l`

This makes sense because there is 1 process at the start, and in this process, 27 extra processes created.

q2.4

Using man, I grep for "stack_size"

`cat apache2.strace | grep stack_size`

Every single stack_size was equal to 0x7ffec0.
I used printf to convert it to decimal with:
`printf "%d\n" 0x7ffec0`

q3.1

About 227.8 seconds
The observation starts at time 1684276335.857725
The first get request for ICS-Logo-for-dark-150x150-1.png is: 1684276563.701339
1684276563.701339 - 1684276335.857725 = 227.843614

Found the times using these two commands:
`head apache2.strace -n 1`
`cat apache2.strace | grep ICS-Logo | head -n 1`

q3.2

32 GET requests

Command used:
`grep ICS-Logo apache2.strace | grep GET -c`

q3.3

32 Total GET requests for the image
`cat apache2.strace | grep ICS-Logo | grep GET -c`

q3.4

It is not corrupted. A good way to check is the png header's first 8 bytes which is expected to be: 137 80 78 71 13 10 26 10

Using the command:
`cat apache2.strace | grep ICS-Logo -A 10 | tail -n 20`
I found the bytes from the `write(...)` system call.
`{iov_base="\x89\x50\x4e\x47\x0d\x0a\x1a\x0a...`
which in decimal turns out to be
`['137', '80', '78', '71', '13', '10', '26', '10']`
which is the same as the expected png header!

q4.1

There is 1 listen syscall, so it seems to listen and wait for a connection once

`cat apache2.strace | awk '{print $3}' | cut -d'(' -f1 | sort | grep listen -c`

q4.2

About 0.505 seconds

I tried to first find the process that was accepting all the requests. I ran `cat apache2.strace | grep accept` and `cat apache2.strace | grep epoll_wait` and found that the process that was accepting requests PID 2071760.

I looked for the clone and fork syscall that created that process and found got the following:
`2071734 1684276336.363042 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7fc5d6ffd910, parent_tid=0x7fc5d6ffd910, exit_signal=0, stack=0x7fc5d67fd000, stack_size=0x7ffec0, tls=0x7fc5d6ffd640} => {parent_tid=[2071760]}, 88) = 2071760 <0.000095>`

The observation start: 1684276335.857725
Time of process creation: 1684276336.363042

So about 0.505 seconds from the start the process was created

q4.3

0.517194 sec between messages

Reading `man epoll_wait`, the syscall returns the number of file descriptors that are ready to be read. So we should only look for when epoll_wait has file descriptors read to know how often the request arrival rate is.

I redirected the times into the file using the piped command:
`cat apache2.strace | grep 2071760 | grep epoll_wait | grep '= 1' | awk '{print $2} > request_times.txt'`

Then I wrote a short c script to calculate the average time between messages

q4.4

20 different ips issuing requests to the server
I grepped for GET requests to the server and then filtered out ips using regex
`cat apache2.strace | grep GET | grep -Eo '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' | sort | uniq | wc -l`

q5.1

About 3% overhead (not writing the json file)

I found the writev that comes shortly after the read. I read man and saw that one of the structs it passes in the amount of bytes to write to a buffer. I grepped for `iov_len` to find that the json takes up 9675 bytes and the http overhead was 293 bytes. 293 bytes takes up about 3% of the total overhead.
`cat apache2.strace | grep 1684276562.314195 -A 5 | grep writev | grep iov_len`

q5.2

It found the size using the newfstatat (or fstatat) system call. The return value is a struct that holds the length of the data from the `stat.st_size` attribute

q.5.3

May 11, 2023

Reading the man page for fstatat, an attribute holds the last modification time which is `stat.st_mtime` or `stat.st_mtim` which gives the last modification time.
By grepping for the `st_mtim` attribute, I find that it was last modified at
`cat apache2.strace | grep 1684276562.314195 -A 5 | grep st_mtim`
Which gave a time of `st_mtime=1683833861`

Finding the date
`date -d @1683833861`
outputs:
`Thu May 11 09:37:41 HST 2023`