System Vs OS Virtualization

Student: Shuaiyu Wang

Environment Setup

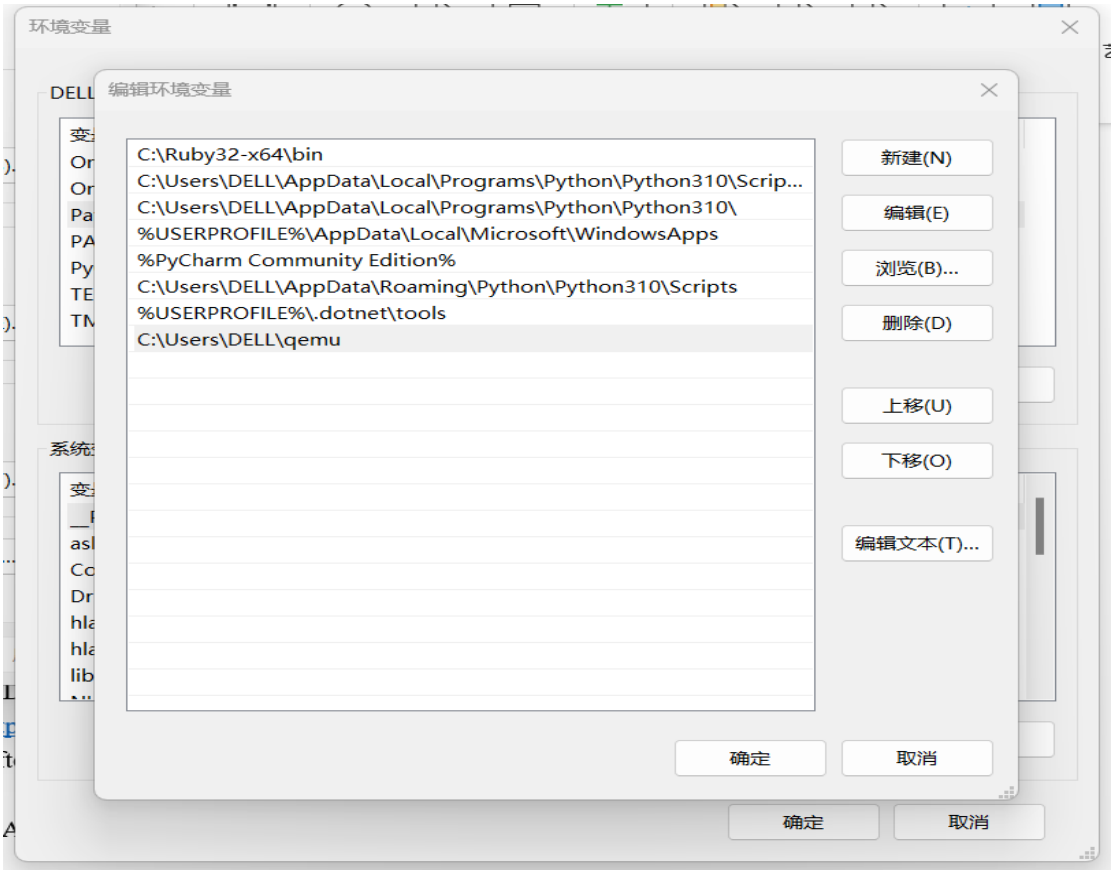
Host System Configuration:
CPU: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz
RAM: 8 GB Storage: 512MB
System Type: 64-bit operating system, x64-based processor
Primary OS: Windows 11

System Virtualization Setup

QEMU Setup
Below are the steps followed to install QEMU on Windows:
1. Downloaded ubuntu server ISO image from the following link:
https://releases.ubuntu.com/16.04/ubuntu-16.04.7-server-amd64.iso

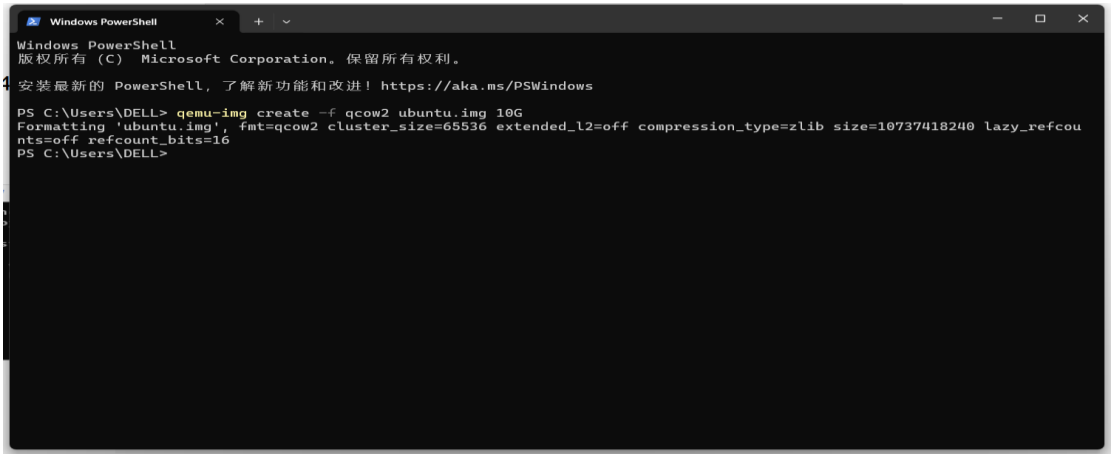
2. Downloaded QEMU for Windows(x64) from the following link:
https://www.qemu.org/download/#windows
After the download is complete, install it.

3. Add QEMU path into Environment Variables as shown in the image below:

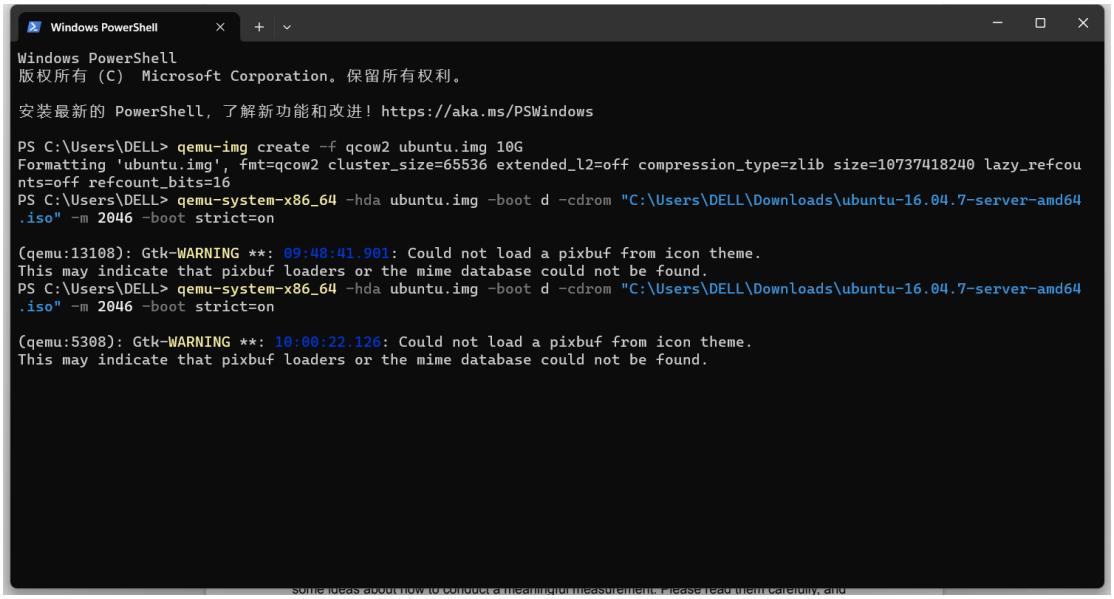


4. Open the command Prompt/Powershell and execute the following command:

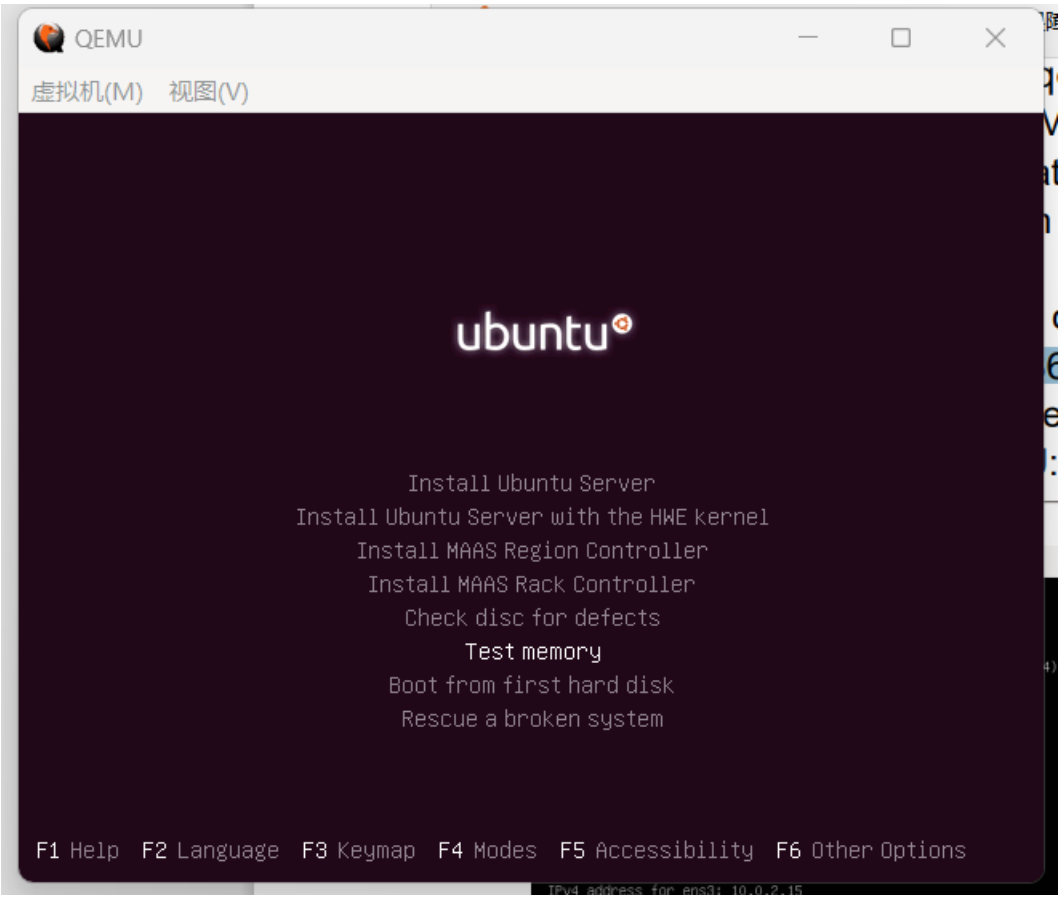
a.Create QEMU image of ubuntu in qcow2 file format using following command:
qemu-img create -f qcow2 ubuntu.img 10G

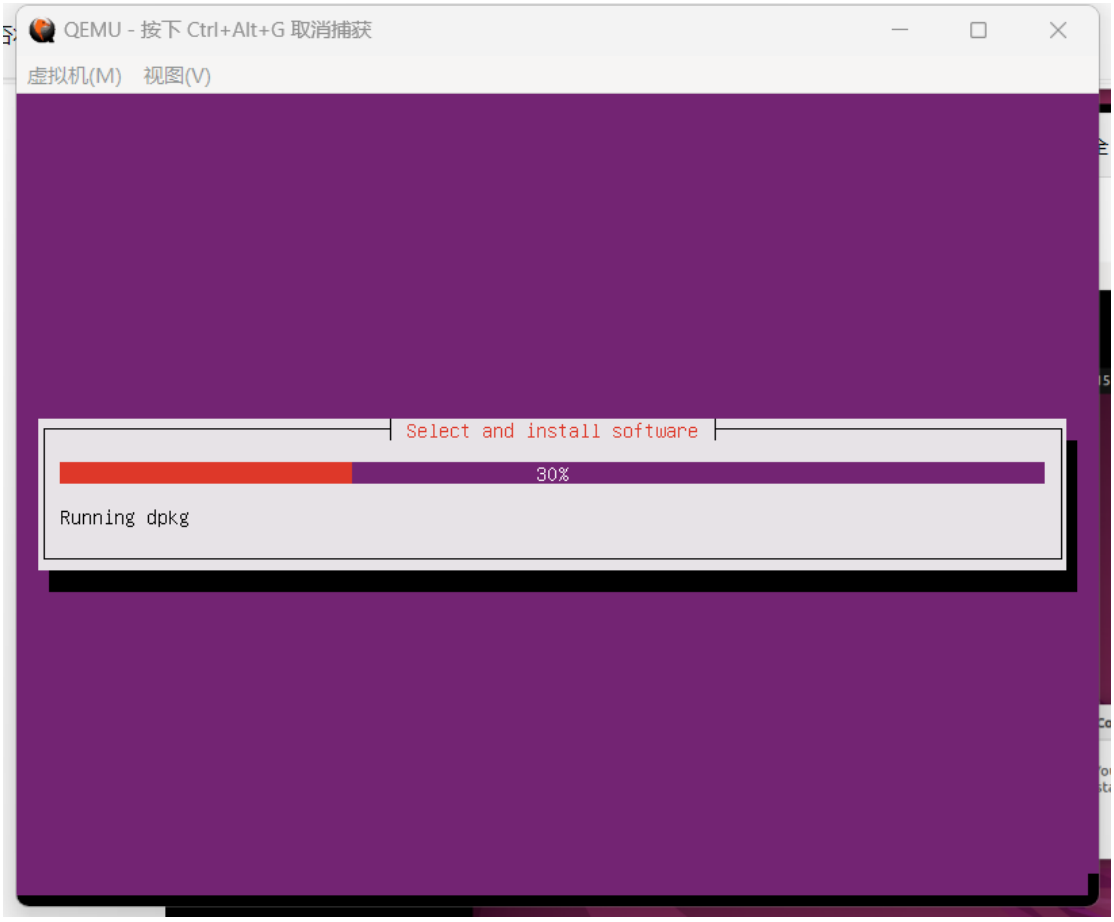


b. Boot iso file on QEMU with the following command:
qemu-system-x86_64 -drive file=ubuntu.img, format=qcow2 -cdrom "C:\Users\DELL\Downloads\ubuntu-16.04.7-server-amd64.iso" -m 2048 -boot d

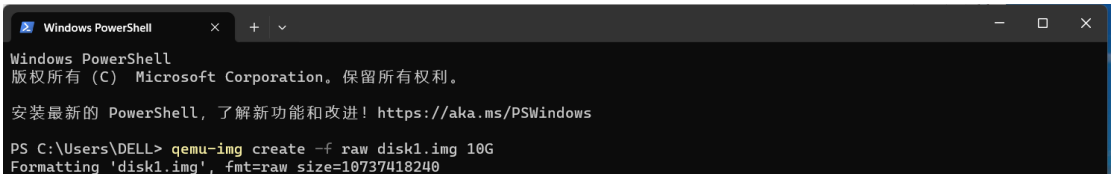


c. The ubuntu now boots up and we do the normal installation as shown in the following images:





d. Then we can create an image in raw format in a similar way:



e. When starting the qemu VM, I used two different smp and two different m values:

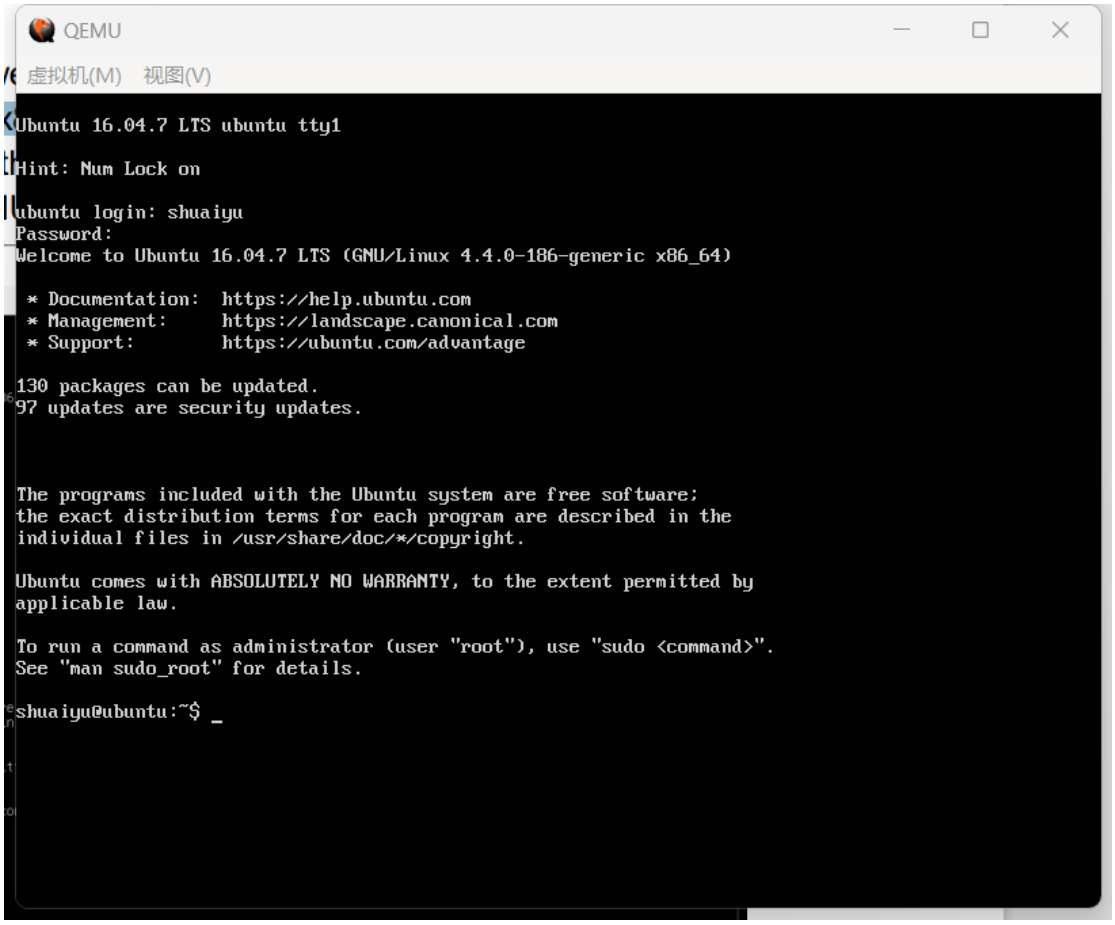
```
(qemu:19212): Gtk-WARNING **: 18:22:45.467: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
PS C:\Users\DELL> qemu-system-x86_64 -drive file=ubuntu.img,format=qcow2 -smp 2 -m 512
```

Use -smp 2 to simulate a dual-core processor, which is useful for scenarios where multi-core processing needs to be simulated, such as testing the behavior of software on multiple cores.
512MB of memory is allocated (via -m 512) in order to test the system's behavior in a resource-constrained environment.

```
PS C:\Users\DELL> qemu-system-x86_64 -drive file=ubuntu.img,format=qcow2 -smp 2 -m 512

(qemu:15892): Gtk-WARNING **: 18:27:17.636: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
PS C:\Users\DELL> qemu-system-x86_64 -drive file=ubuntu.img,format=qcow2 -smp 4 -m 4096

(qemu:17920): Gtk-WARNING **: 18:30:24.825: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
```

This time -smp 4 is used to simulate a quad-core processor, suitable for simulating more powerful

computing capabilities, such as running applications with higher memory or CPU requirements. Using -m 4096 allocates 4GB of memory to provide sufficient resources for running large applications or more complex operating systems.

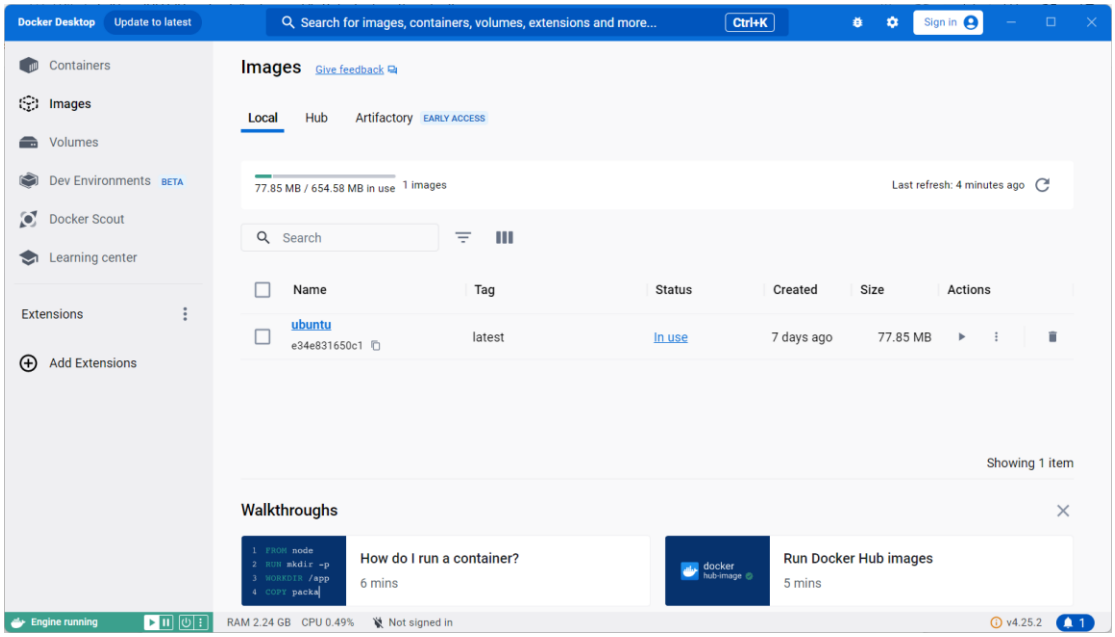


5. Following commands installs sysbench on QEMU
sudo apt-get update
sudo apt install sysbench

Operating System Virtualization Setup

Docker Setup:

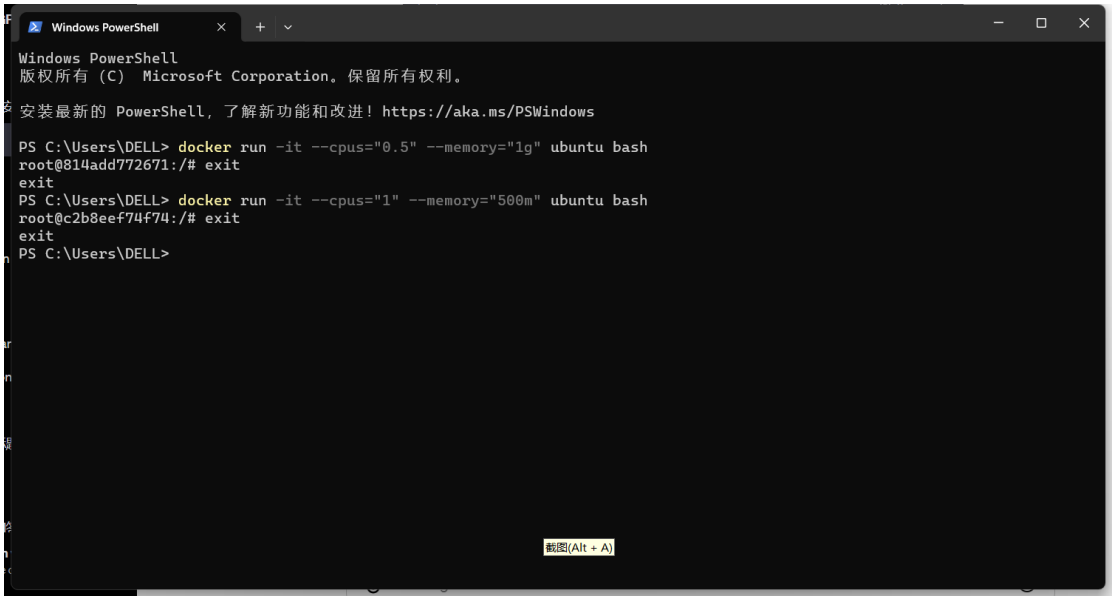
a. Create and start a new container based on Ubuntu image:
docker run ubuntu
b. Start a bash shell in the Ubuntu container:
docker run -it ubuntu bash



The above screenshot shows the image id highlighted of the Ubuntu image that we created and started the bash script.
Image ID: e34e831650c1bb0be9b6f61c6755749cb8ea2053ba91c6cda27fded9e089811f

c. Now, to get the sudo command working we run the following commands:
apt-get update
apt-get -y install sudo
Following command installs sysbench:
sudo apt install -y sysbench

d. When starting a bash shell in the Ubuntu container, I used two different parameters for cpu and memory for experimentation



I used the --cpus="0.5" parameter to test the performance of the container when it was restricted to

using only half a CPU core. The reference group is also --cpu="1", which is very useful for understanding how to allocate resources in a multi-tenant environment to ensure that no one container takes up too much CPU time and affects other containers.

In this experiment, I used the --memory="500m" parameter to limit the container's memory usage to 500M. At the same time, the reference group is --memory="1g", which allows you to test the behavior of the container under memory-limited conditions and how it handles insufficient memory. This is very helpful for memory management optimization of applications.

Proof of Experiment

Docker Running Environment:

1.
```
PS C:\Users\DELL> docker run -it --cpus="1" --memory="500m" ubuntu bash
```
2.
```
PS C:\Users\DELL> docker run -it --cpus="0.5" --memory="500m" ubuntu bash
```
3.
```
PS C:\Users\DELL> docker run -it --cpus="0.5" --memory="1g" ubuntu bash
```
4.
```
PS C:\Users\DELL> docker run -it --cpus="1" --memory="1g" ubuntu bash
```

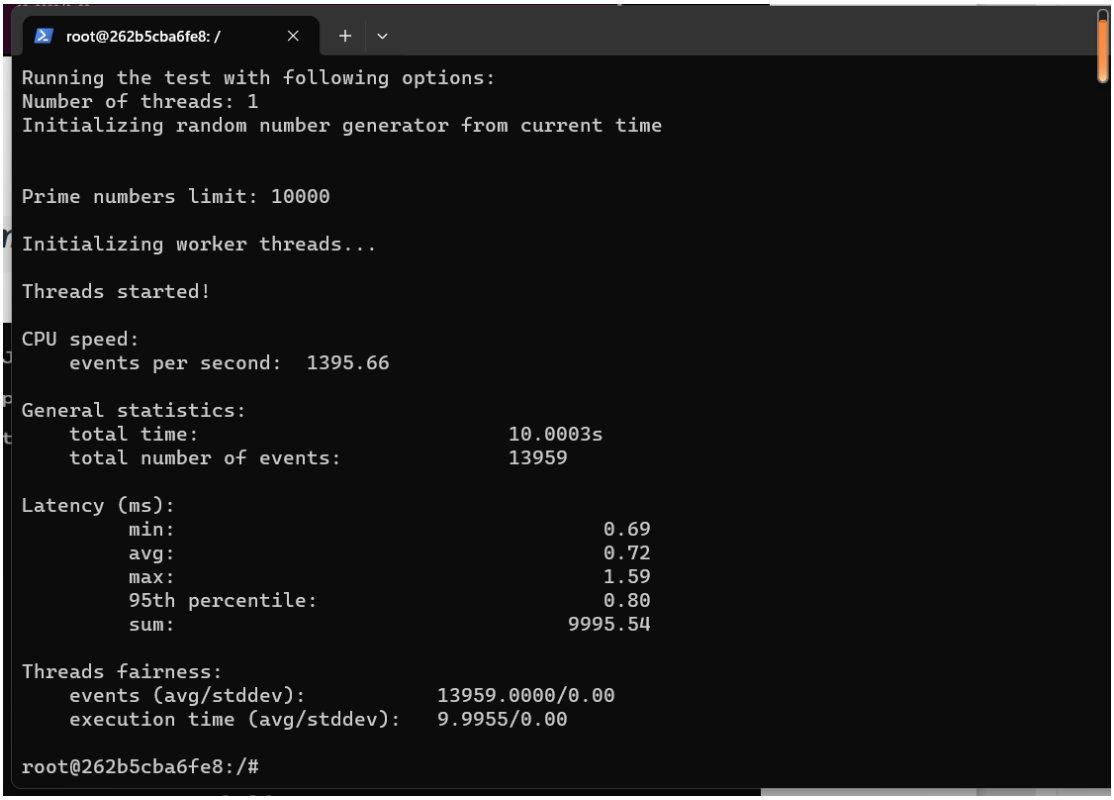
Experiment using sysbench

Environment 1:

1. CPU test:

   a. Test 1:

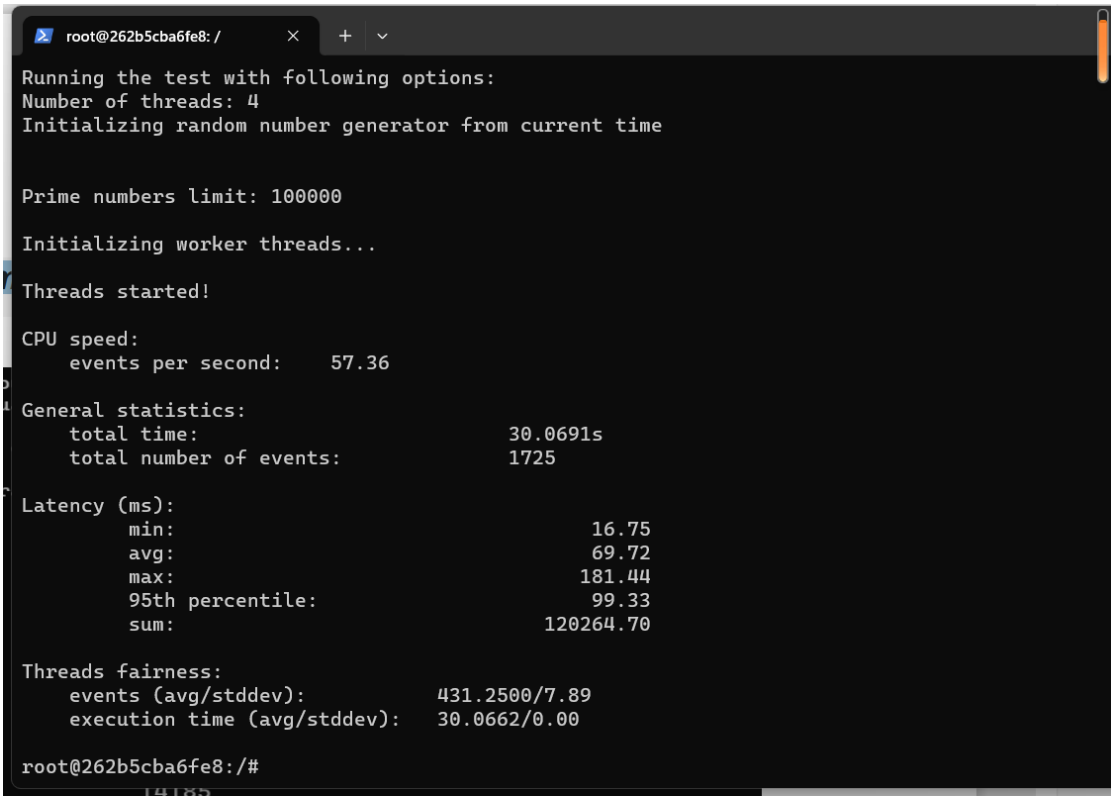
   sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run



b. Test 2:

sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run

```
root@262b5cba6fe8:/# sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 100000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:    57.36

General statistics:
    total time:                          30.0691s
    total number of events:              1725

Latency (ms):
         min:                                    16.75
         avg:                                    69.72
         max:                                   181.44
         95th percentile:                        99.33
         sum:                                 120264.70

Threads fairness:
    events (avg/stddev):           431.2500/7.89
```



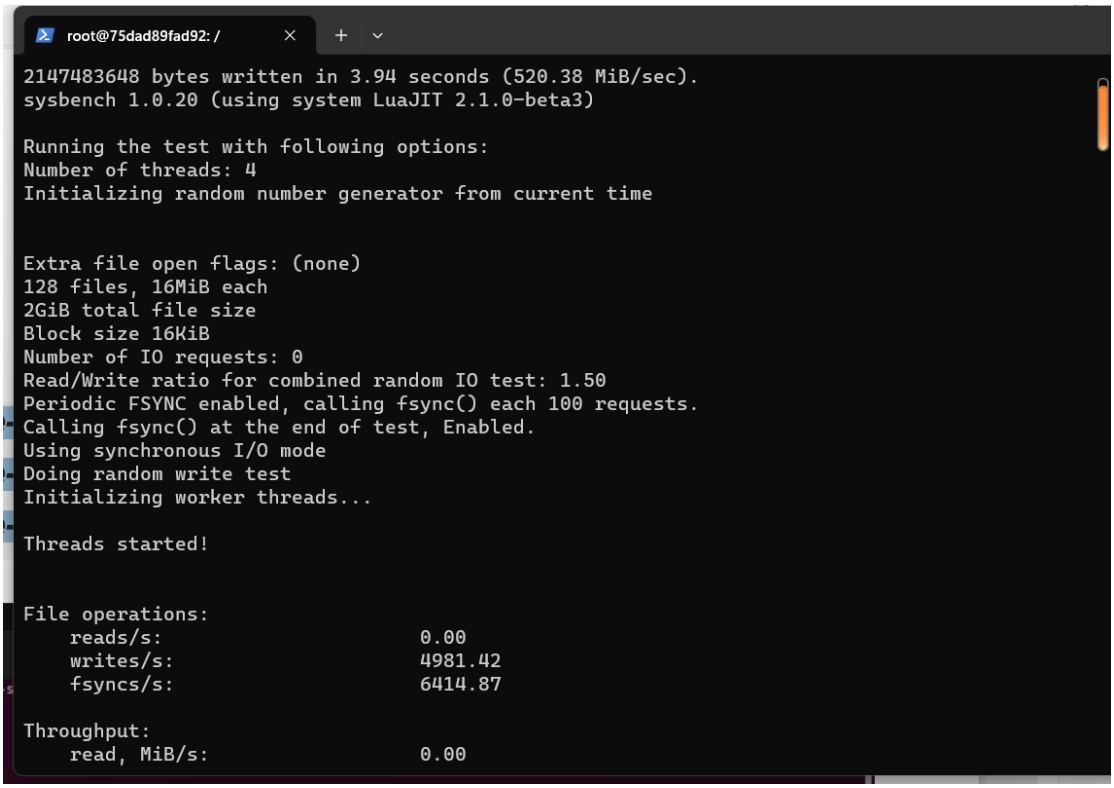
2. FILE I/O test:

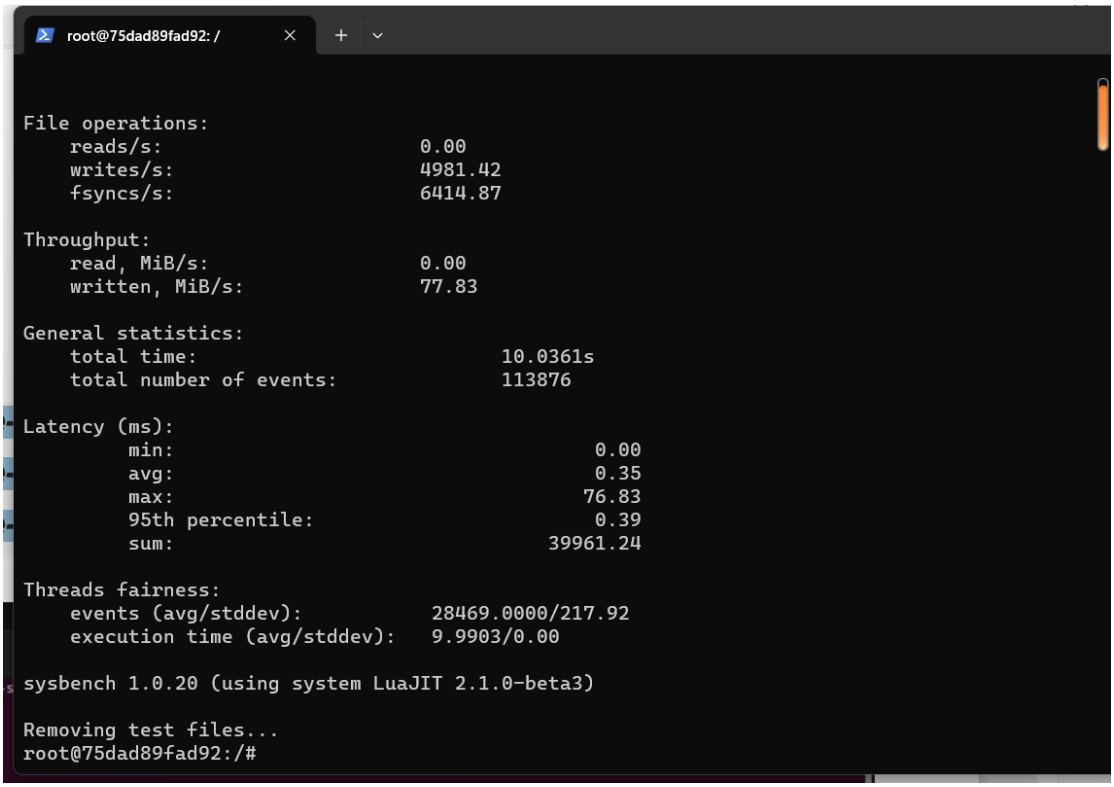
   a. Test 1:

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



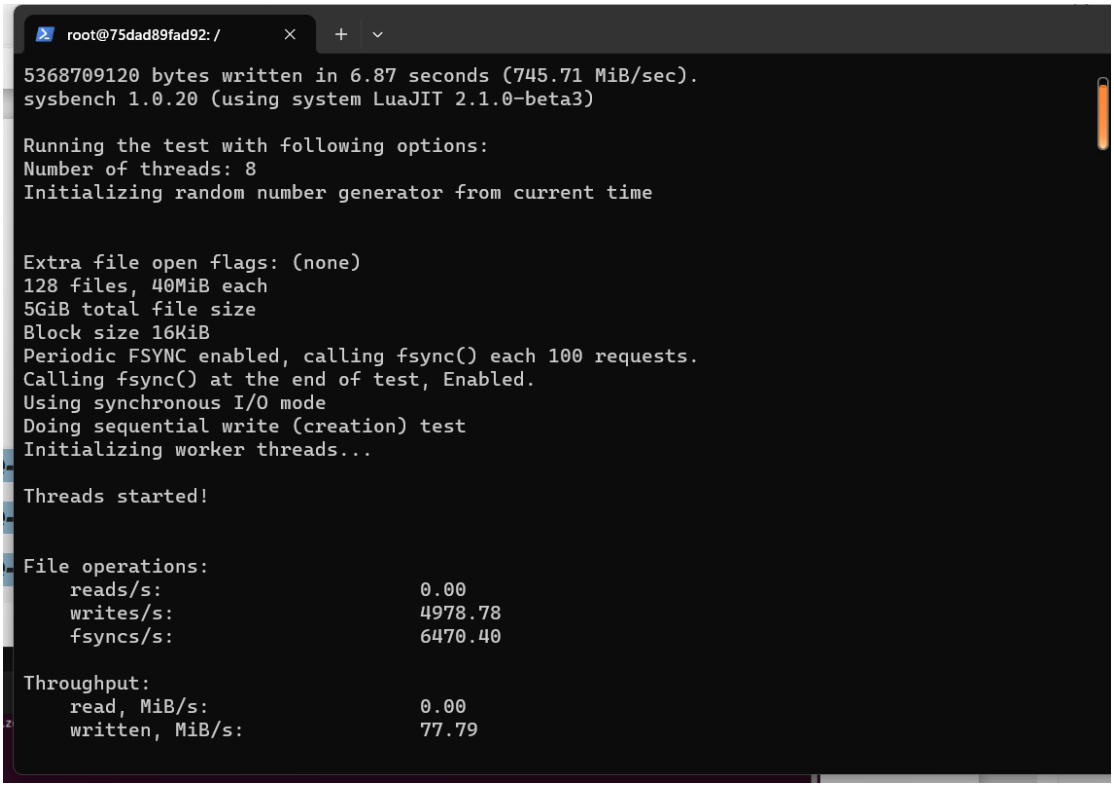


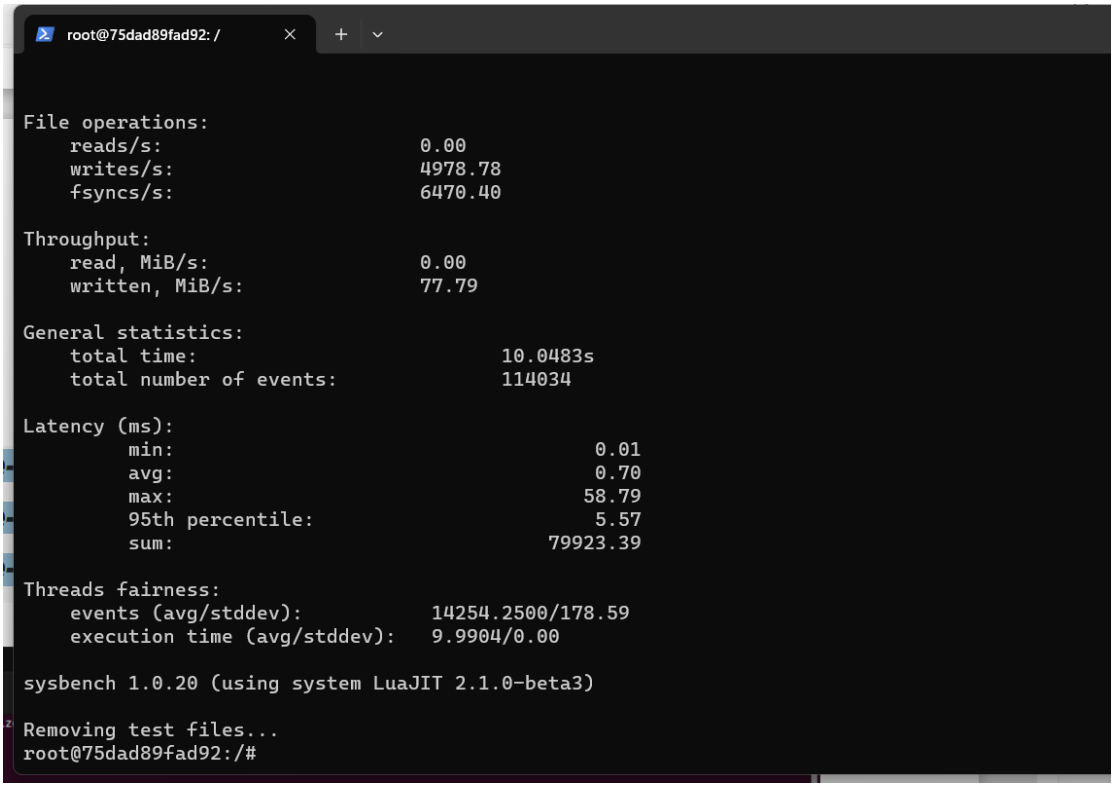
b. Test 2:

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr prepare

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run

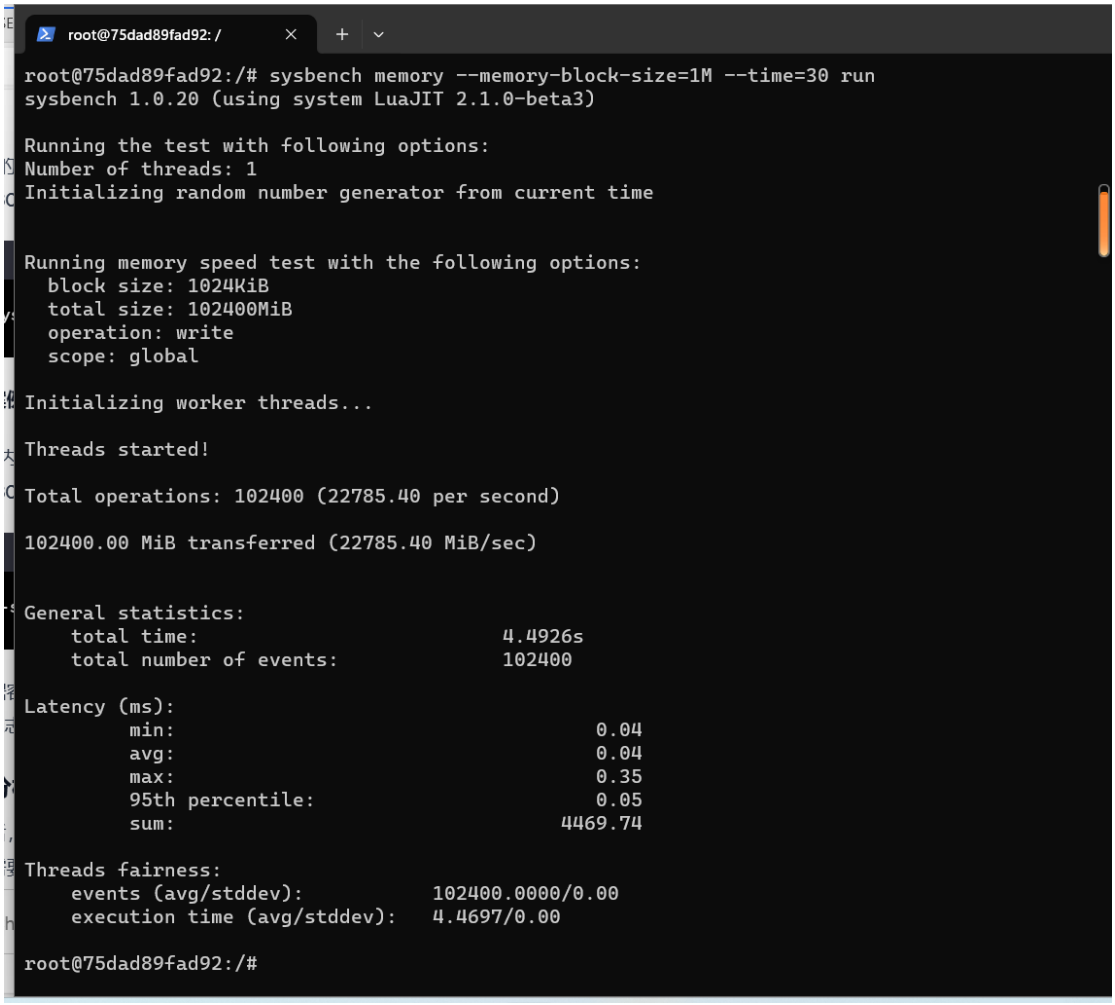
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr cleanup





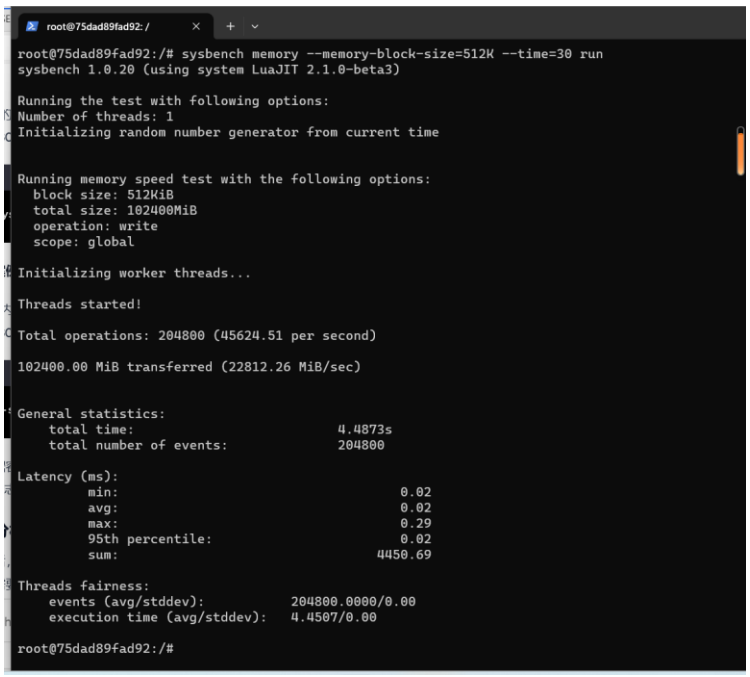
3. Memory test:
   a. Test 1:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 2:

sysbench memory --memory-block-size=512K --time=30 run

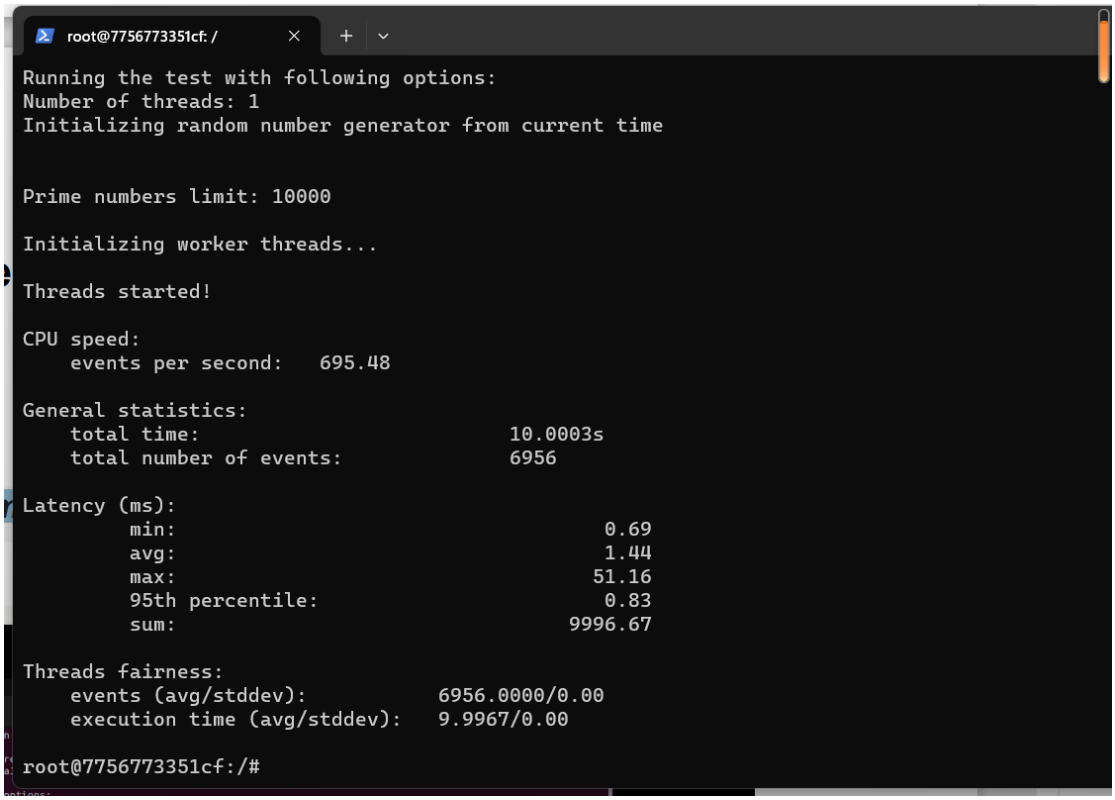


Environment 2:

1. CPU test:

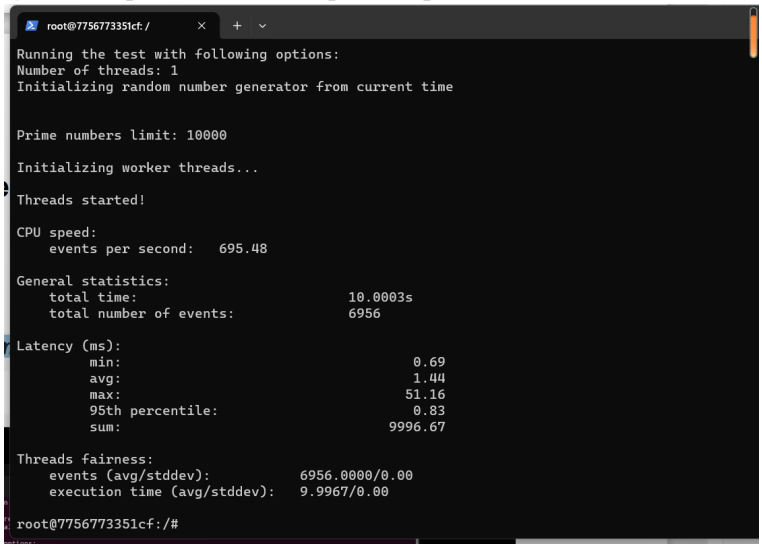
a. Test 3:

sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run



b. Test 4:

sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run



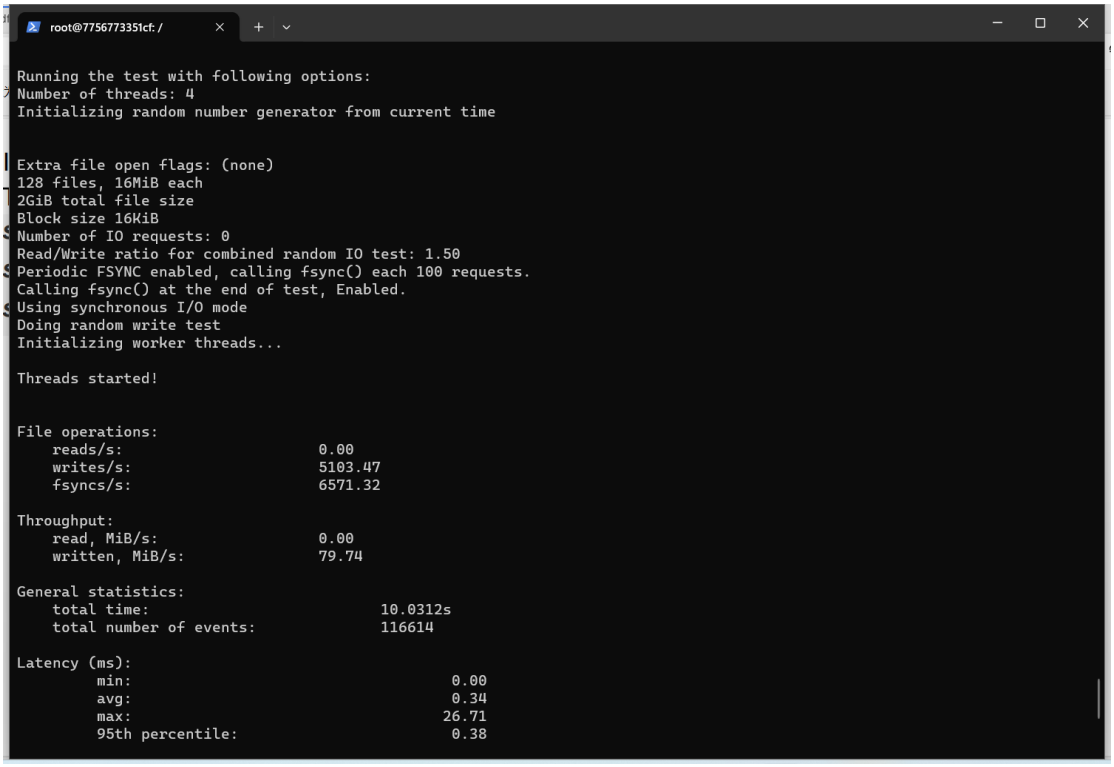
2. FILE I/O test:

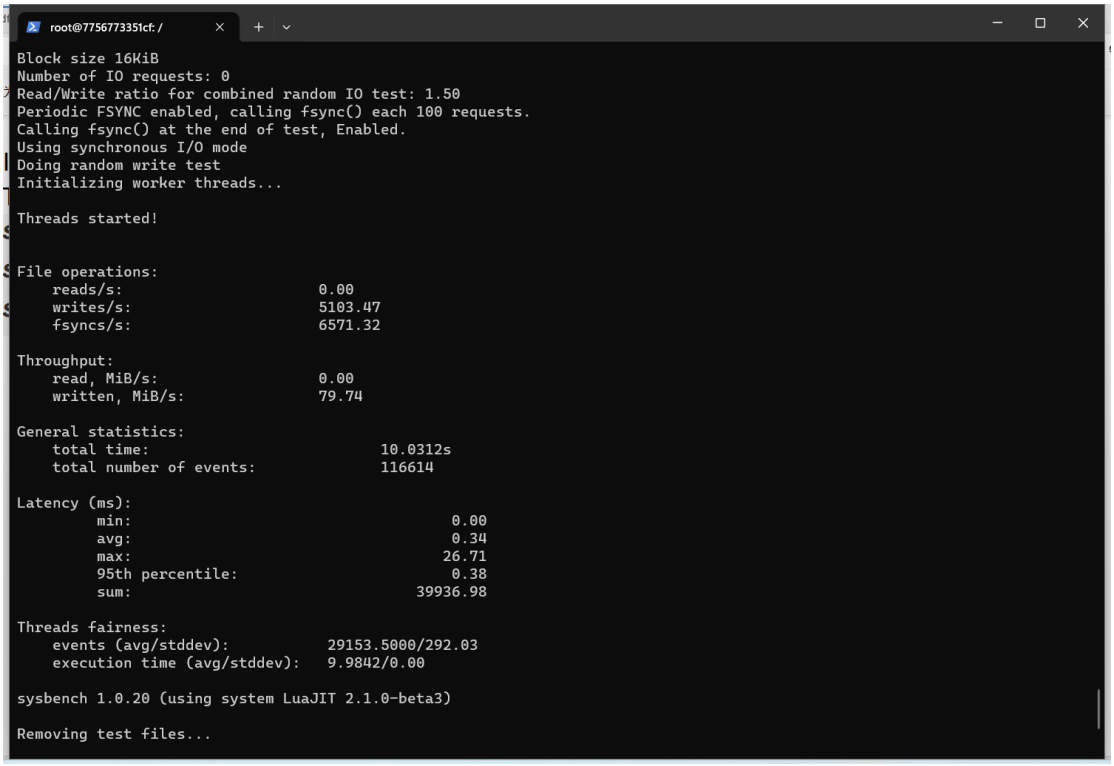
a. Test 3:

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



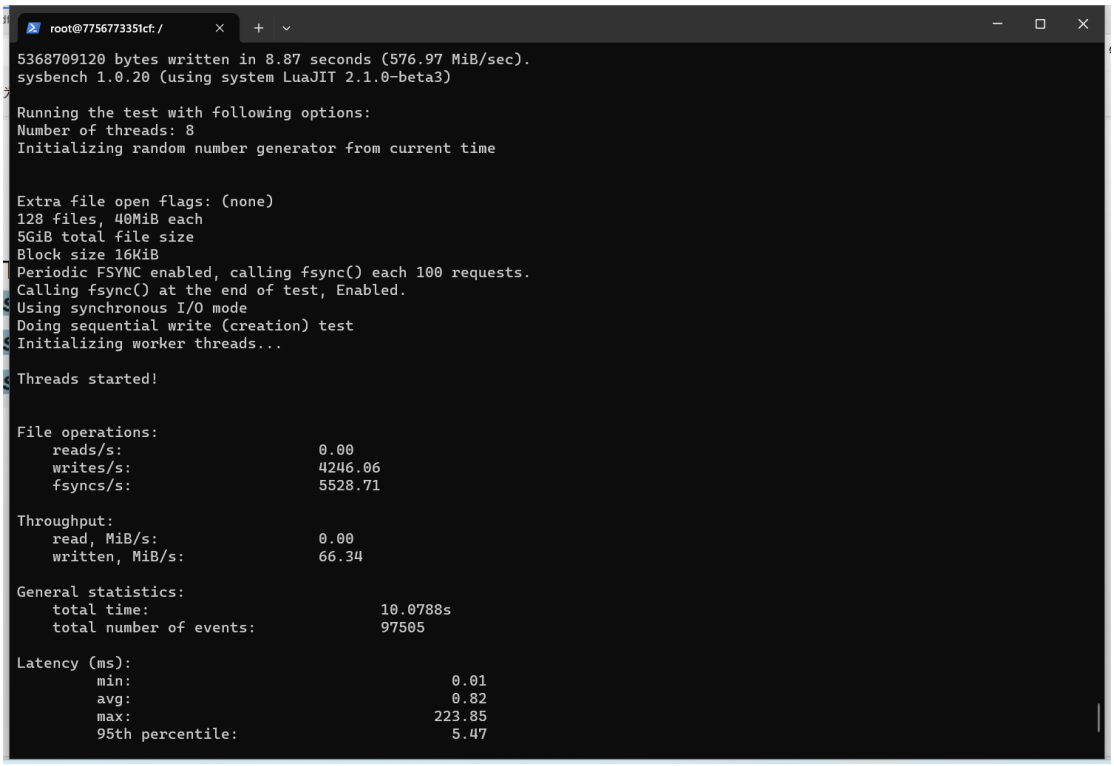


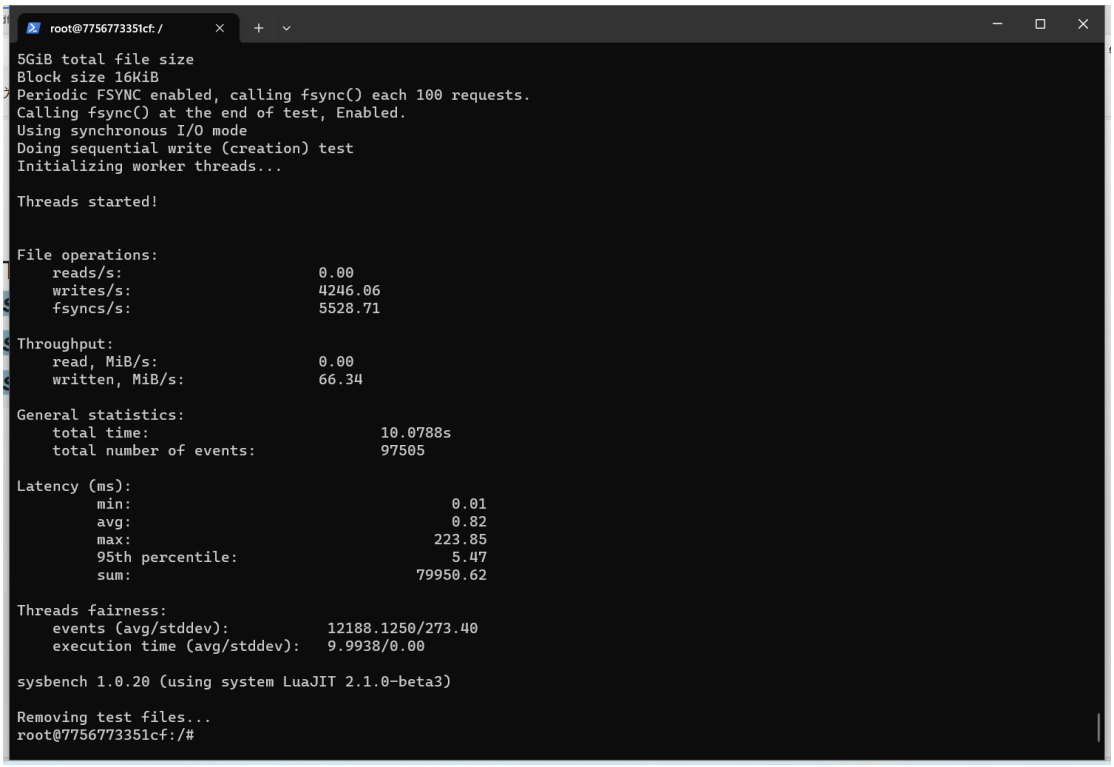
b. Test 4:

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr prepare

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr cleanup

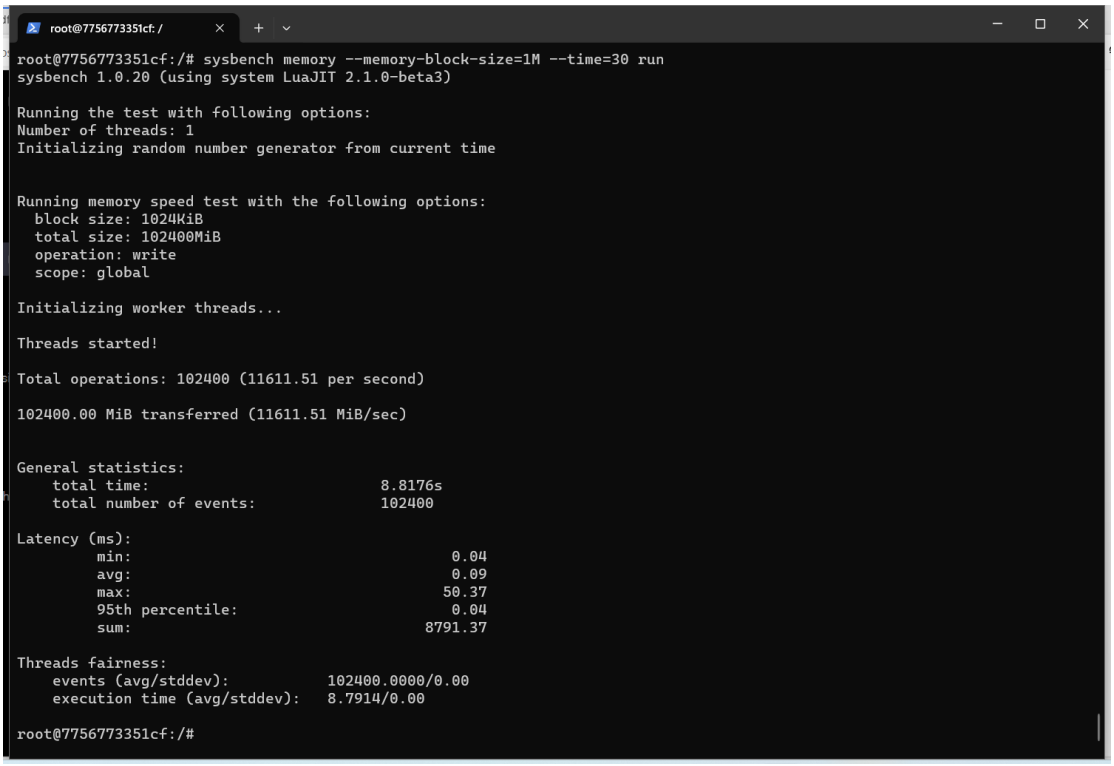




3. Memory test:

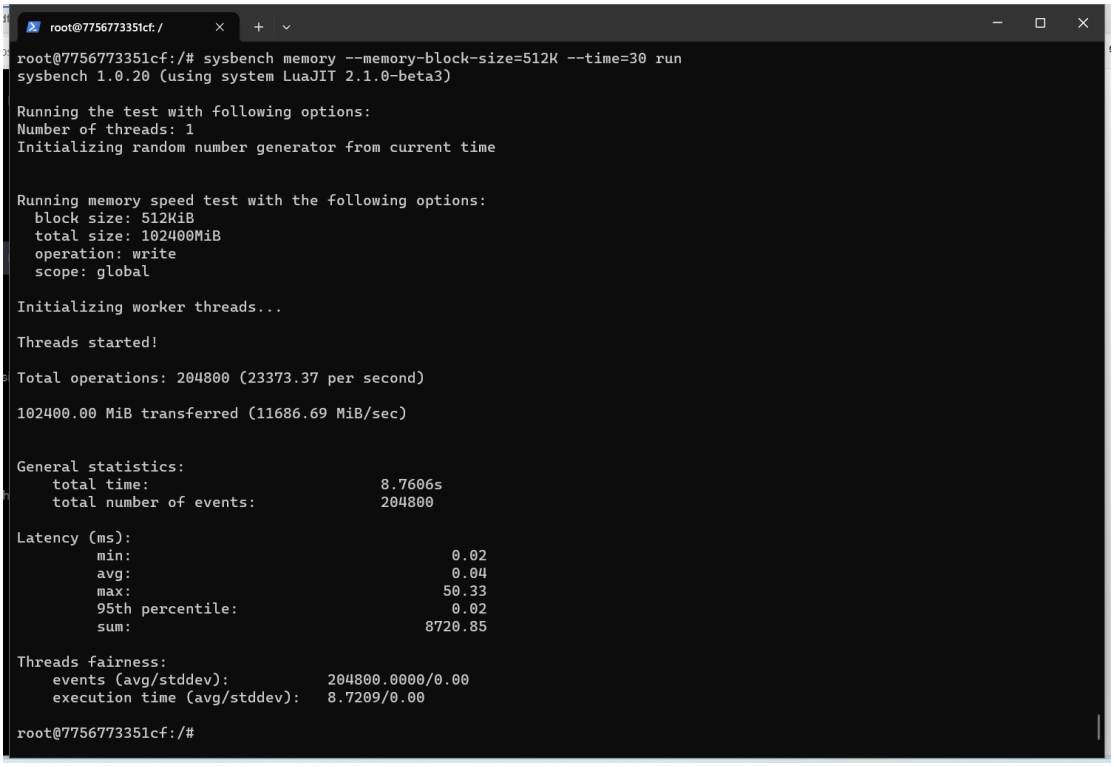
c. Test 3:

sysbench memory --memory-block-size=1M --time=30 run



d. Test 4:

sysbench memory --memory-block-size=512K --time=30 run

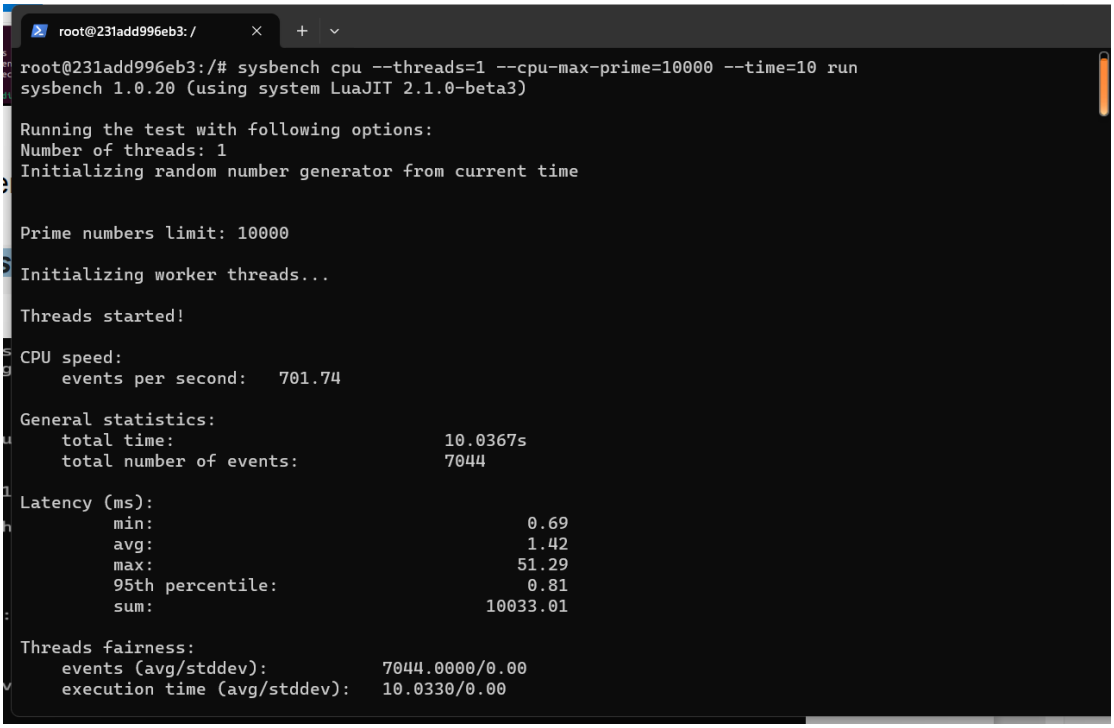


Environment 3:

1. CPU test:

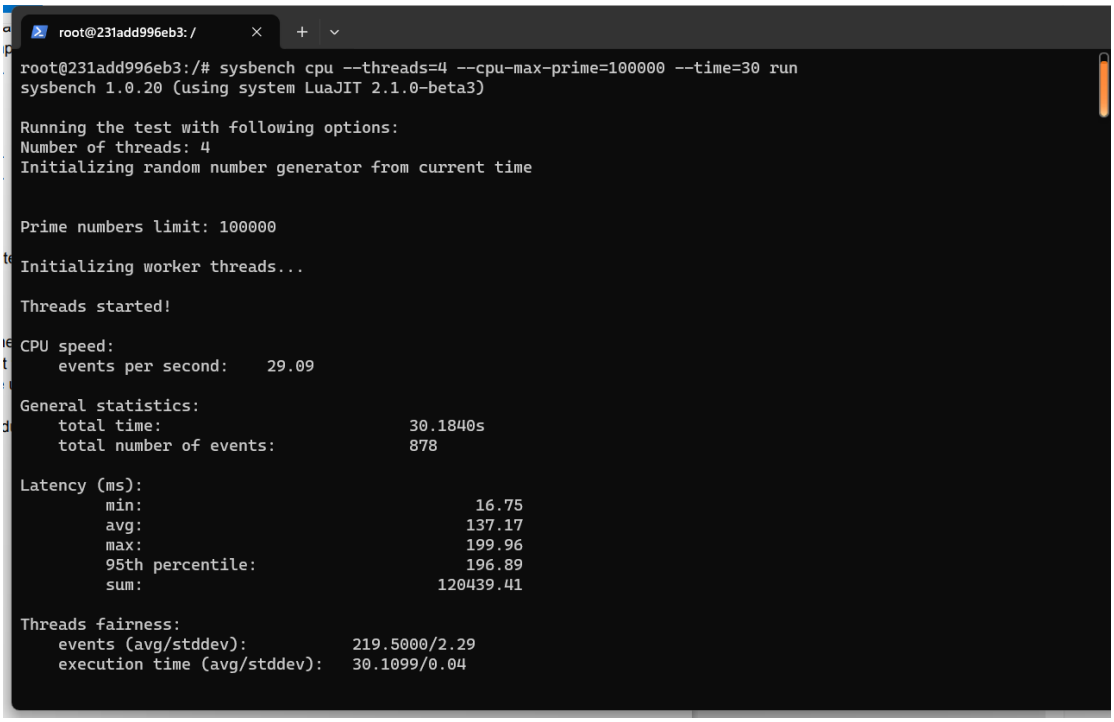
a. Test 5:

sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run



b. Test 6:

sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run



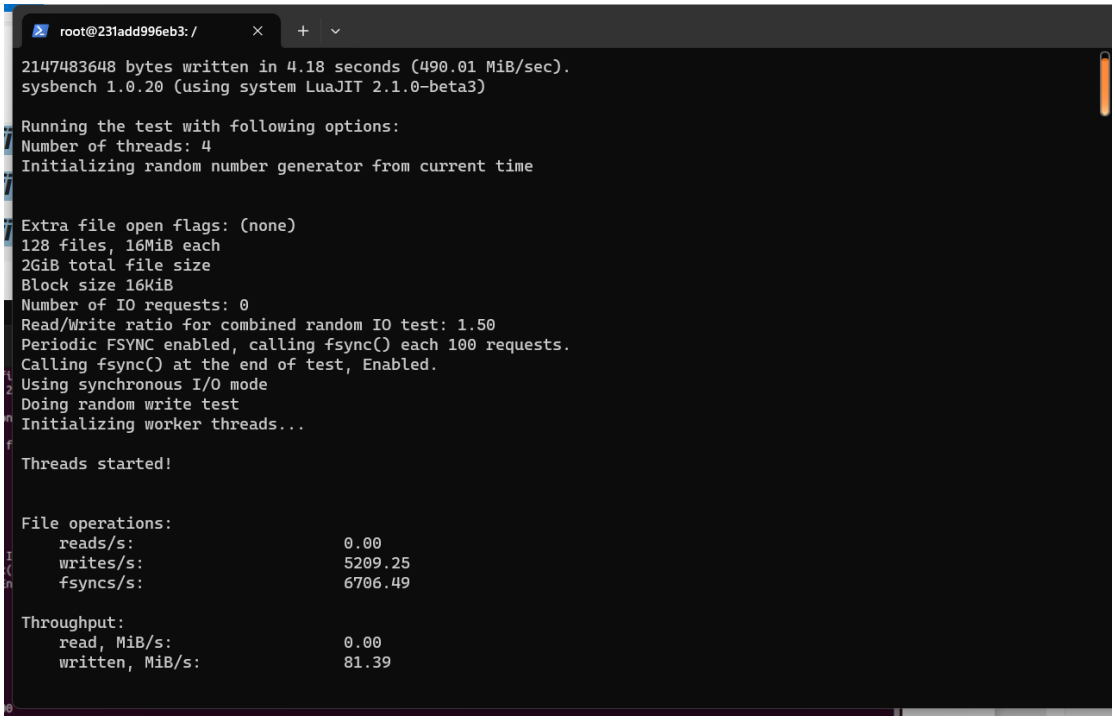
2. FILE I/O test:

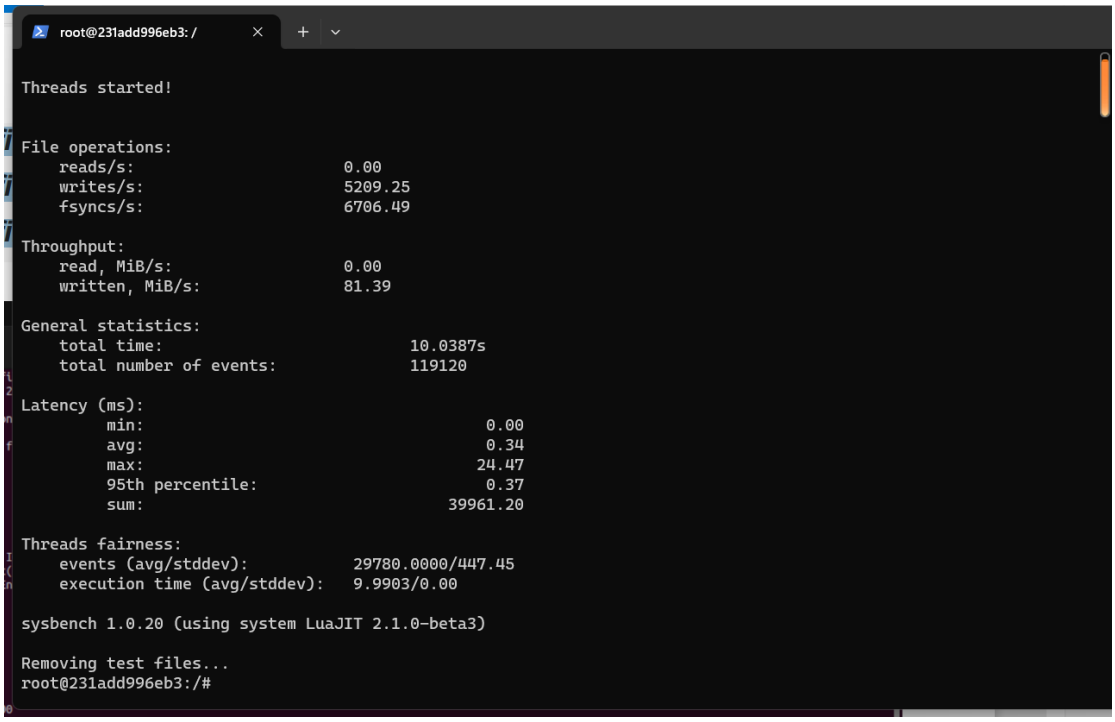
a. Test 5:

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



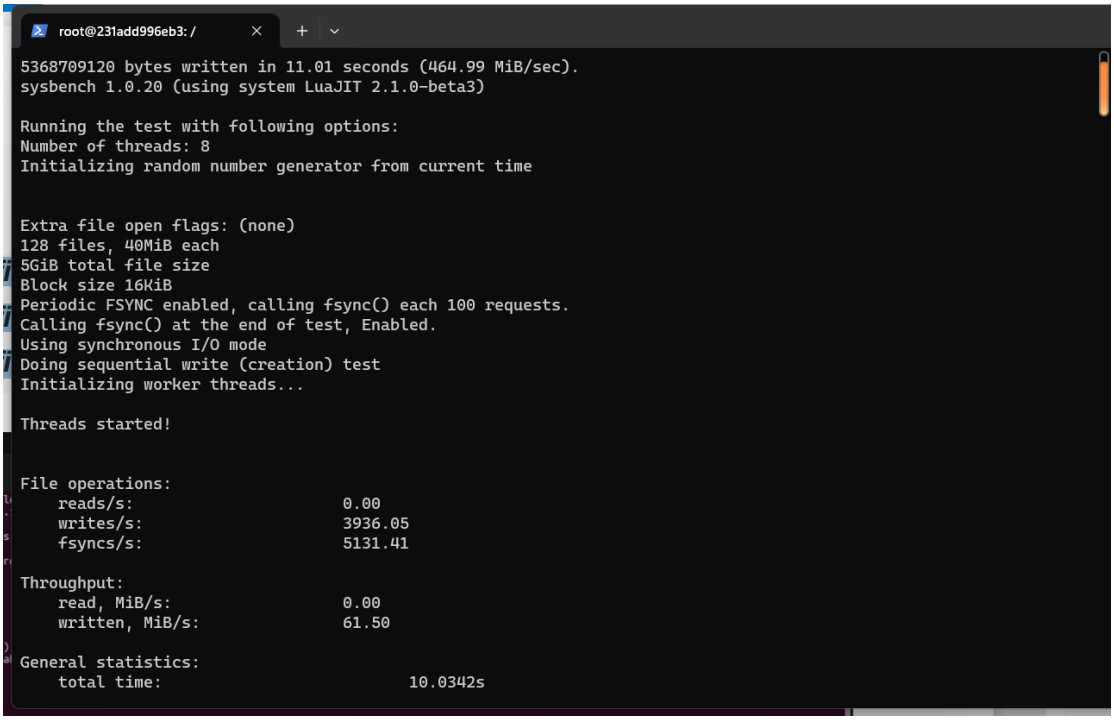


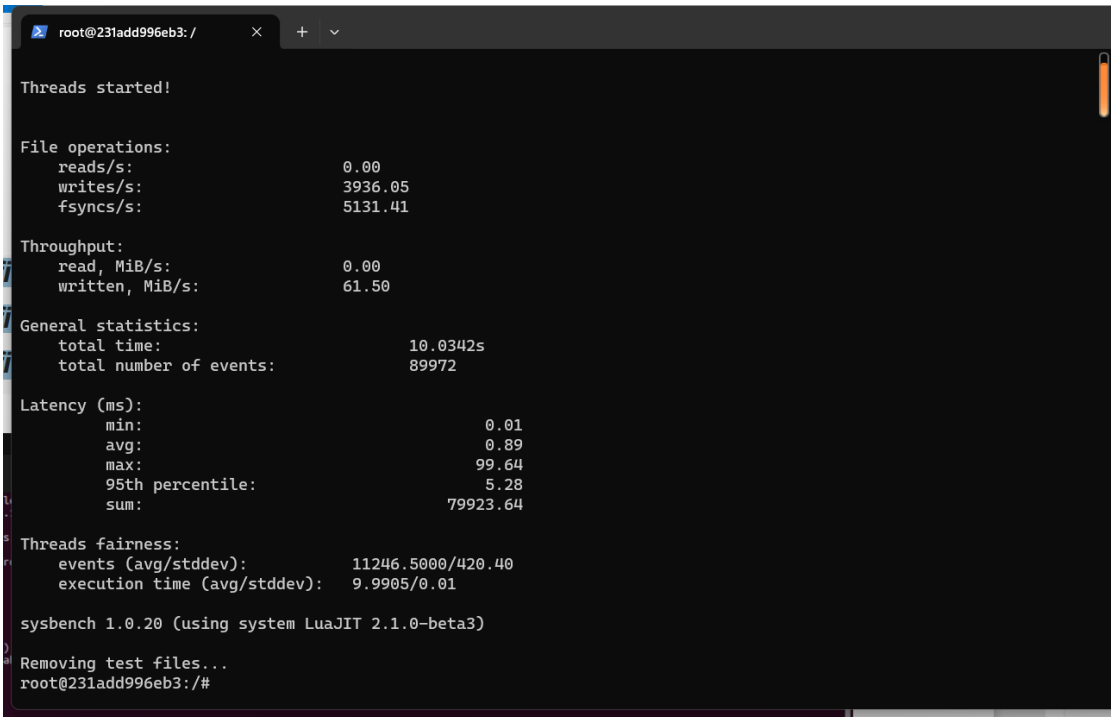
b. Test 6:

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr prepare

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr cleanup

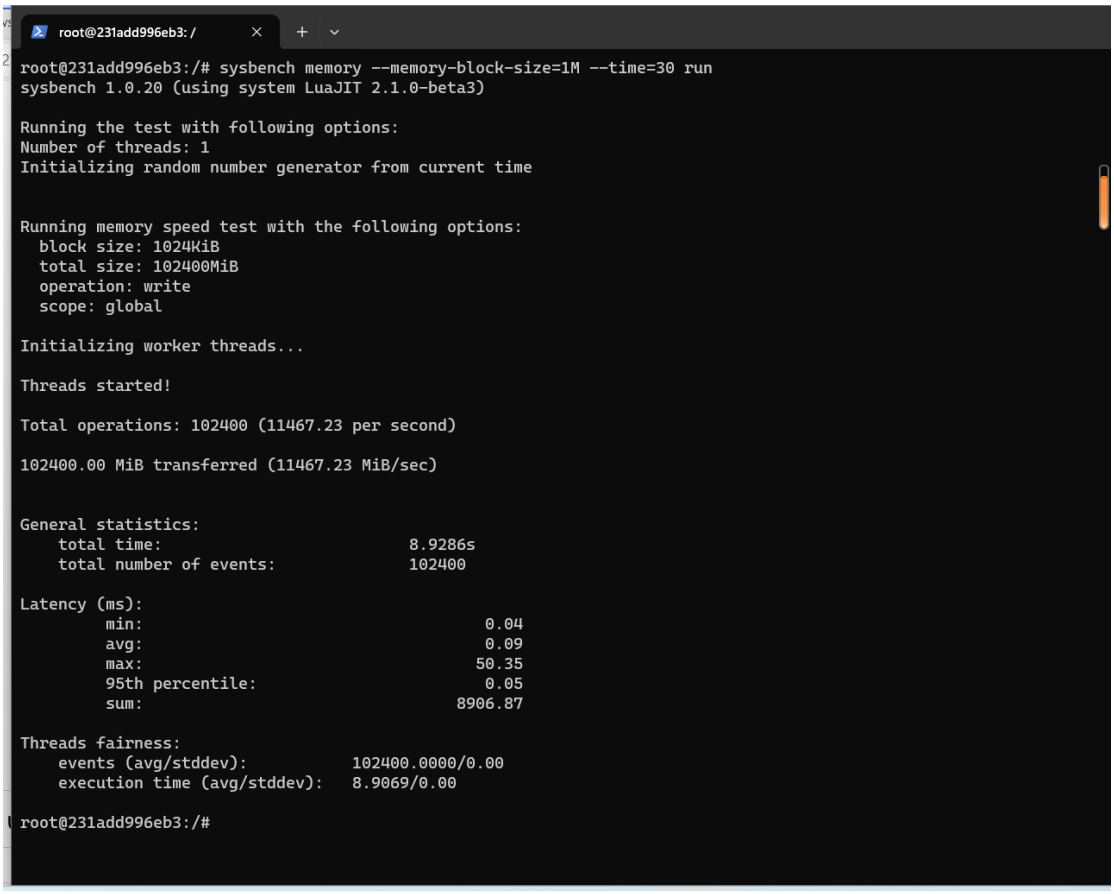




3. Memory test:

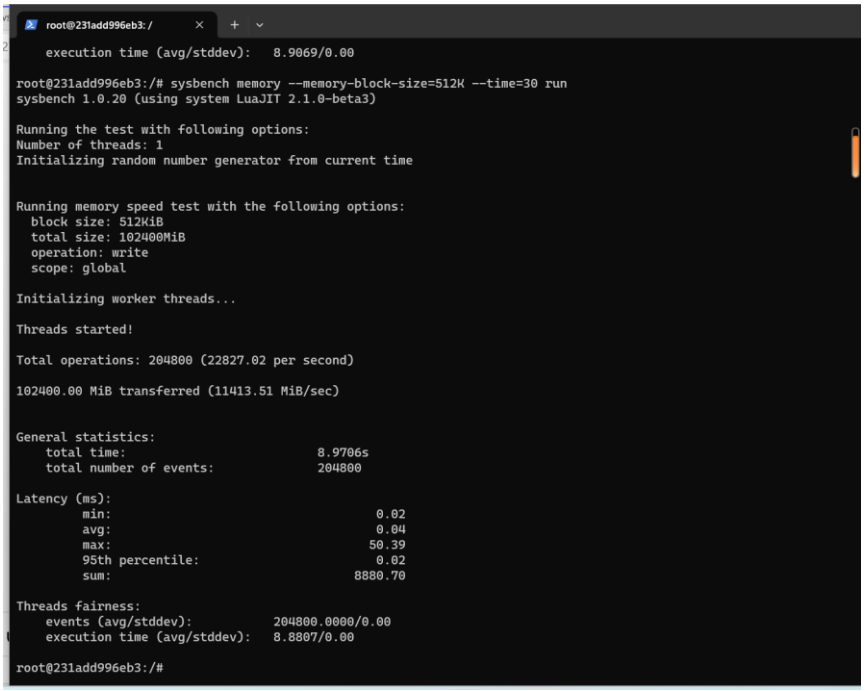
a. Test 5:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 6:

sysbench memory --memory-block-size=512K --time=30 run

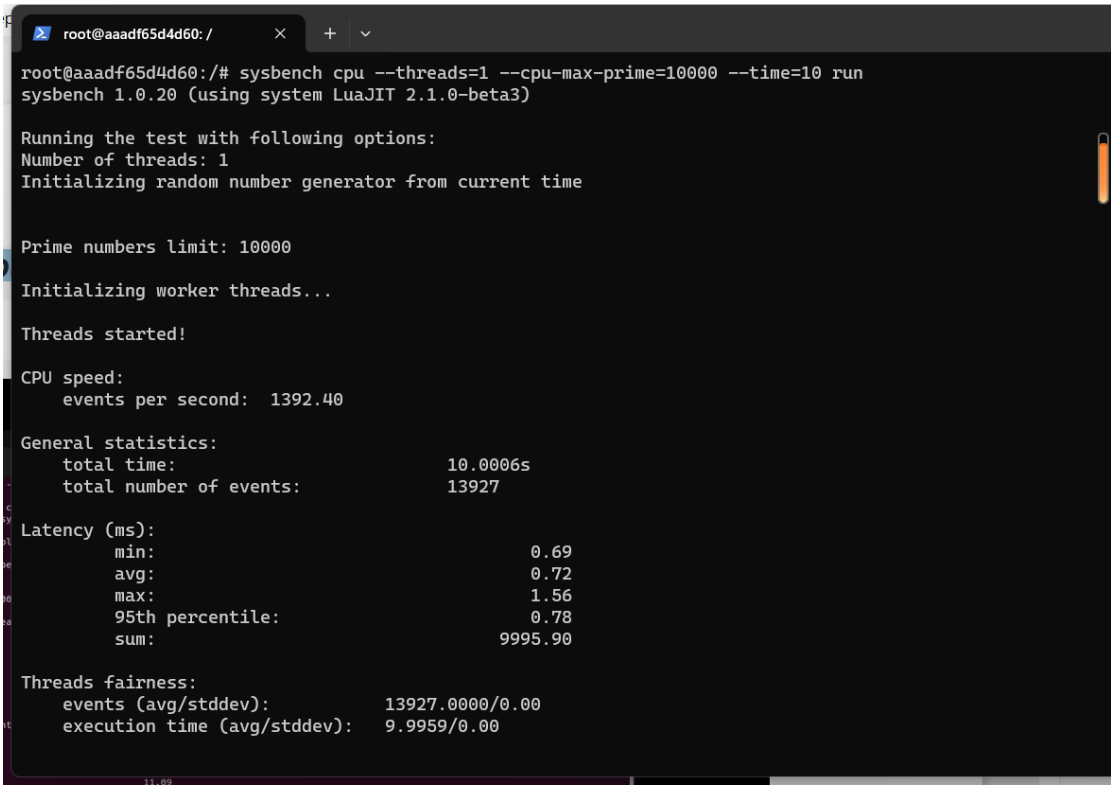


Environment 4:

1. CPU test:

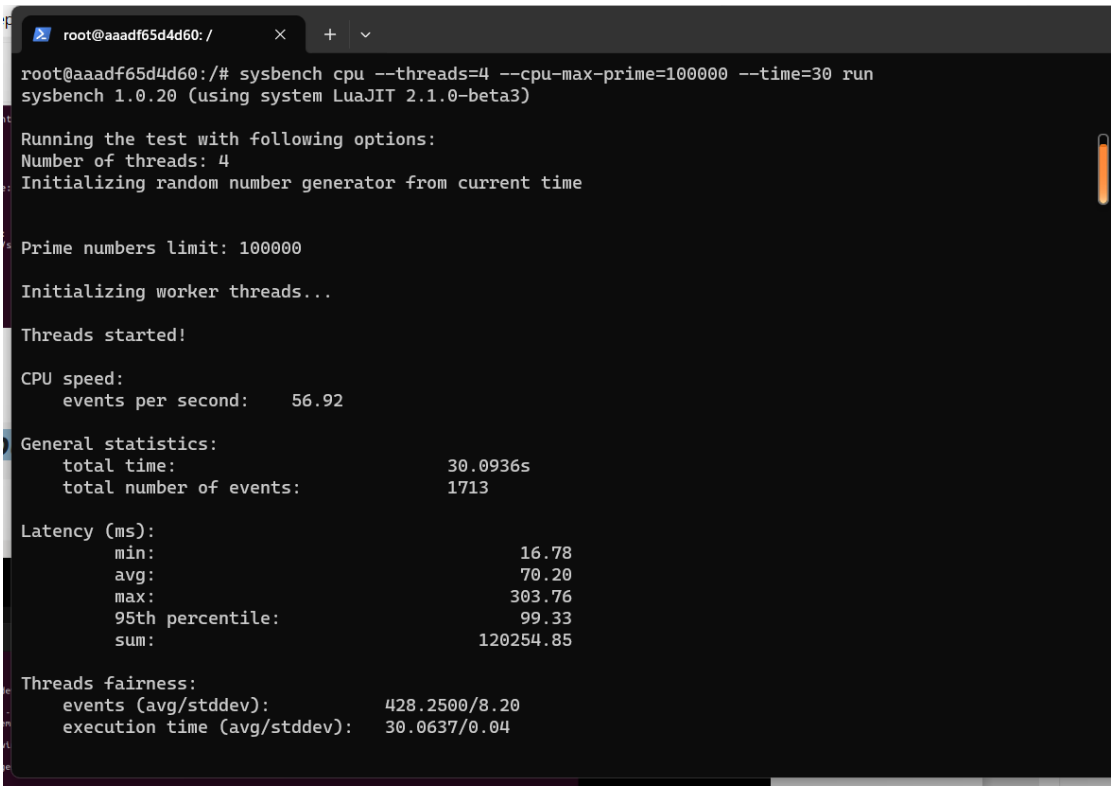
a. Test 7:

sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run



b. Test 8:

sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run



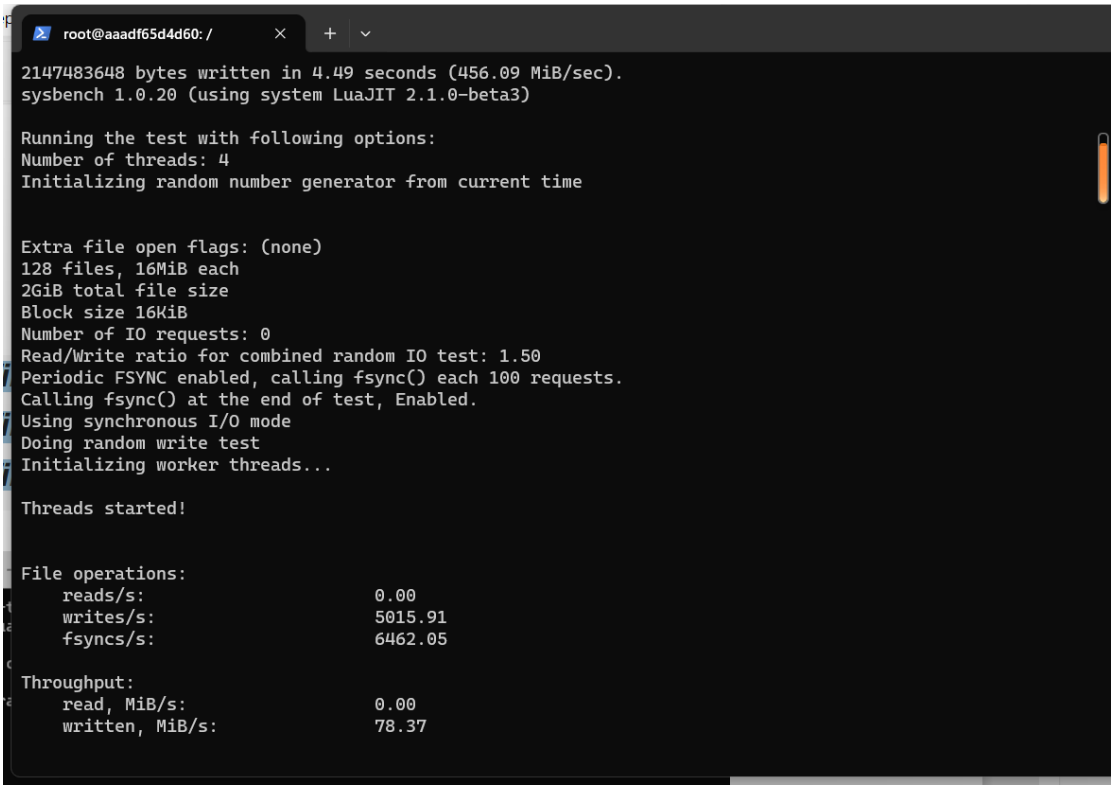
2. FILE I/O test:

a. Test 7:

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup

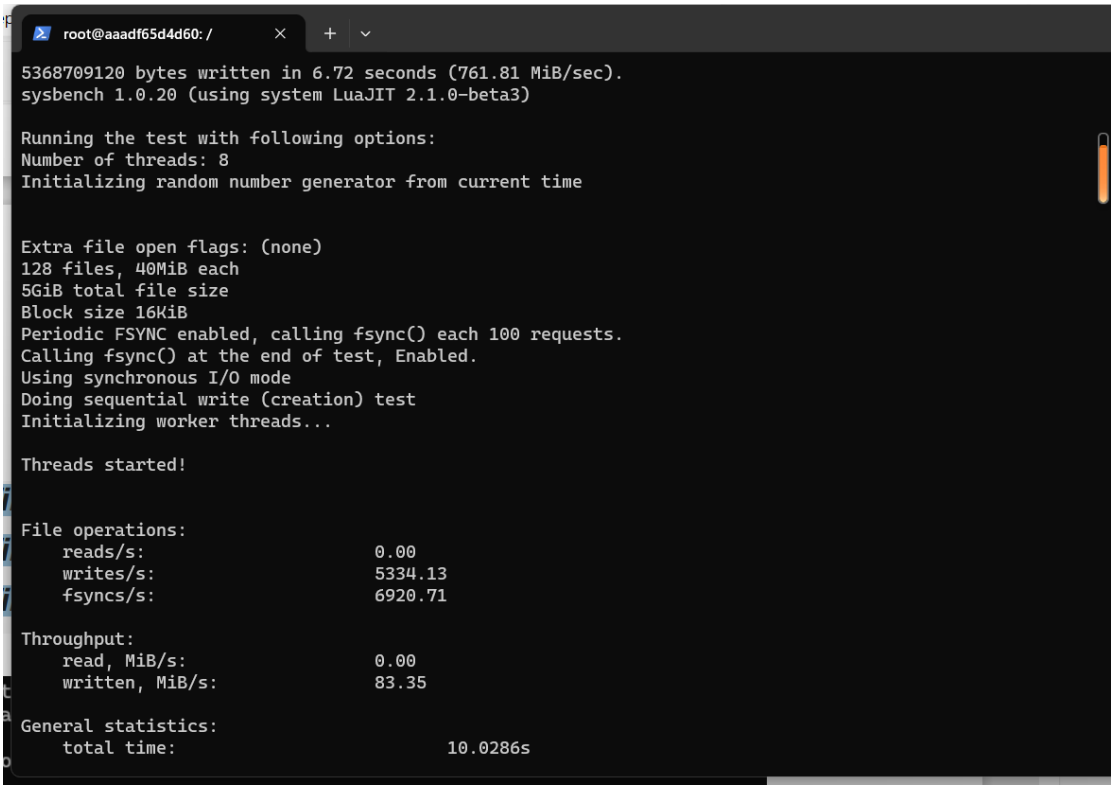


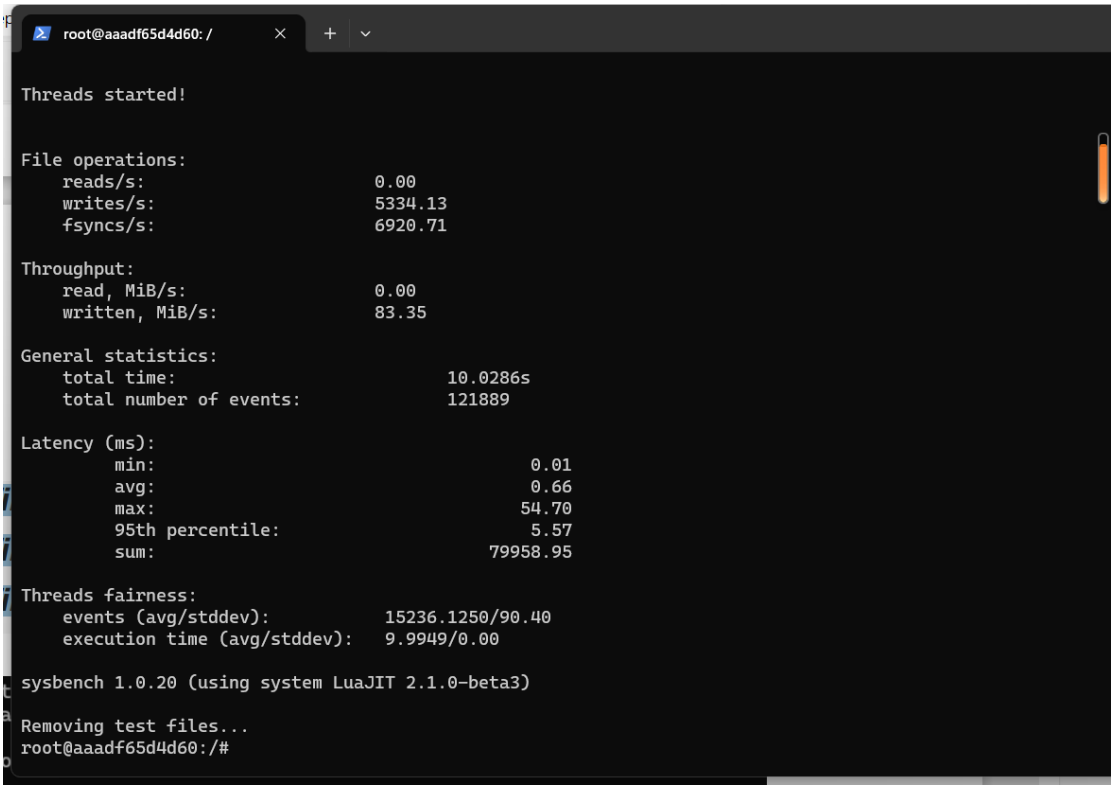
b. Test 8:

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr prepare

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr cleanup

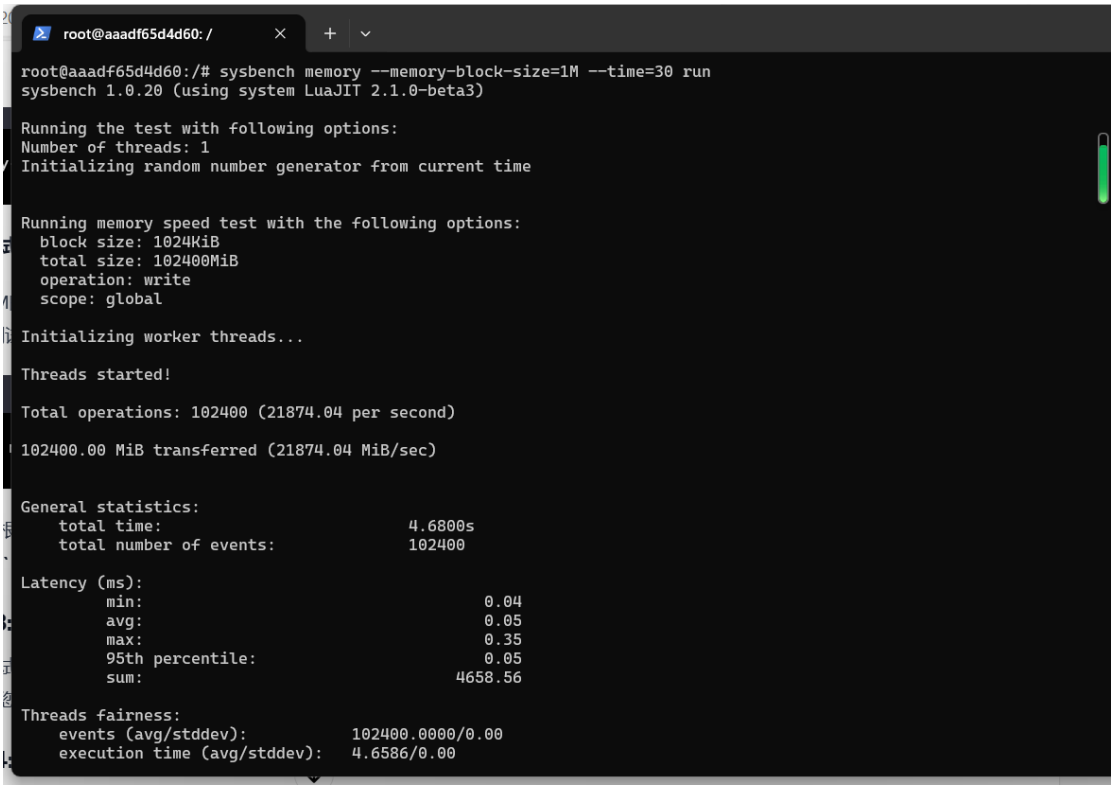




3. Memory test:

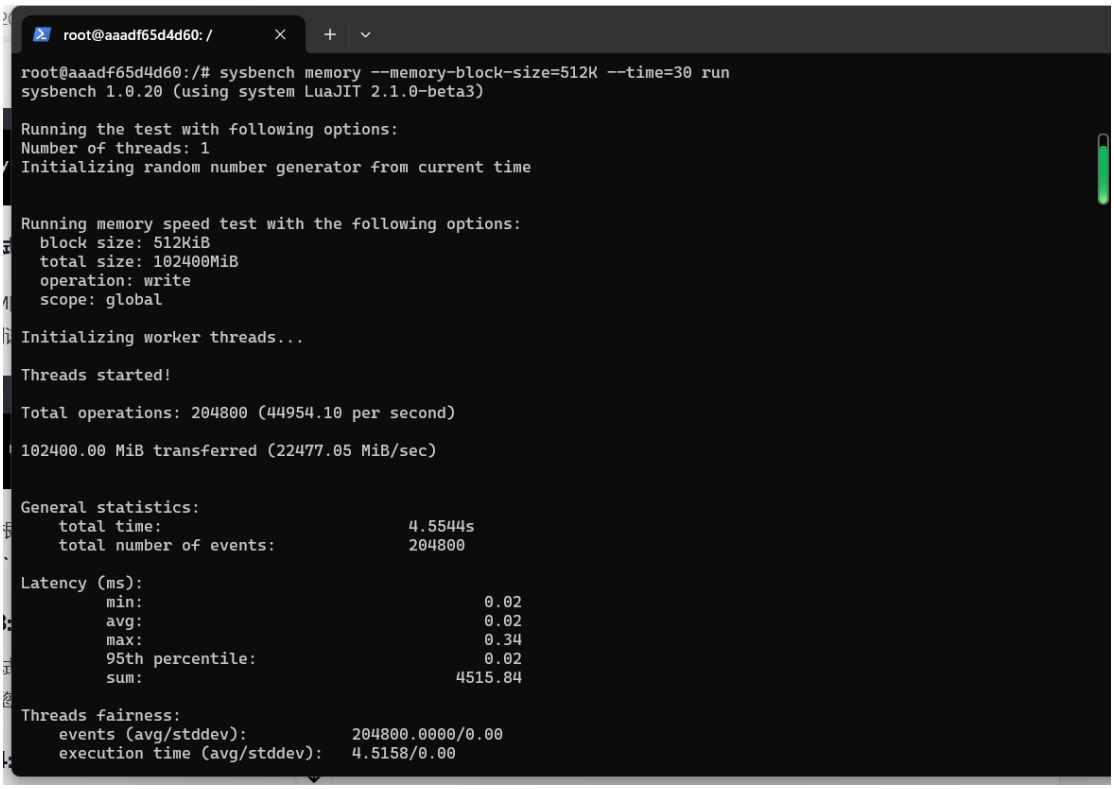
a. Test 7:

sysbench memory --memory-block-size=1M --time=30 run



c. Test 8:

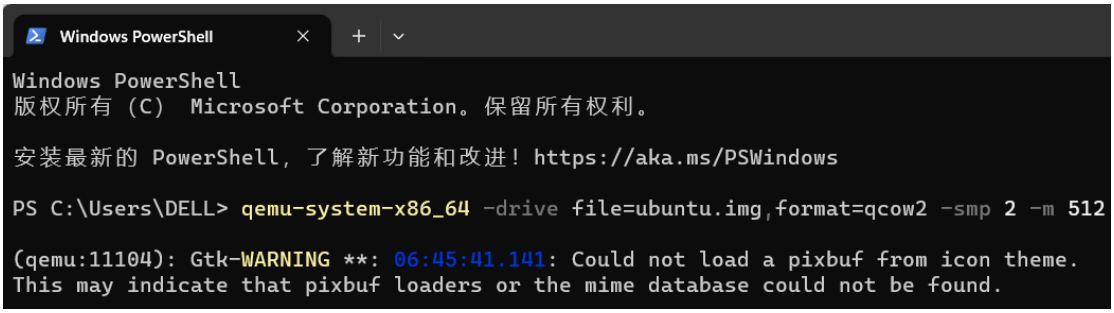
sysbench memory --memory-block-size=512K --time=30 run



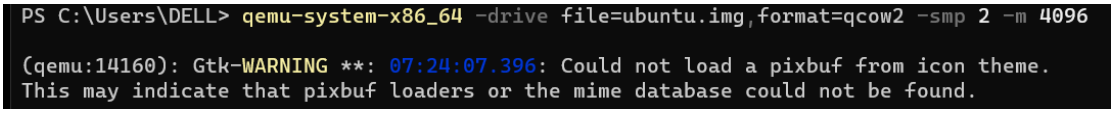
QEMU VM Running Environment:

Image 1: ubuntu.image, format: qcow2

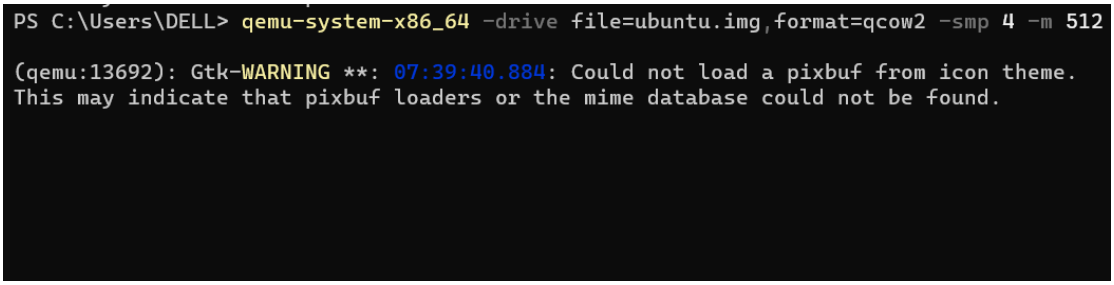
Environment 1:



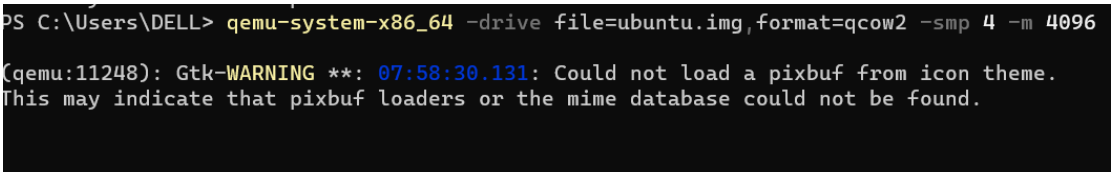
Environment 2:



Environment 3:



Environment 4:



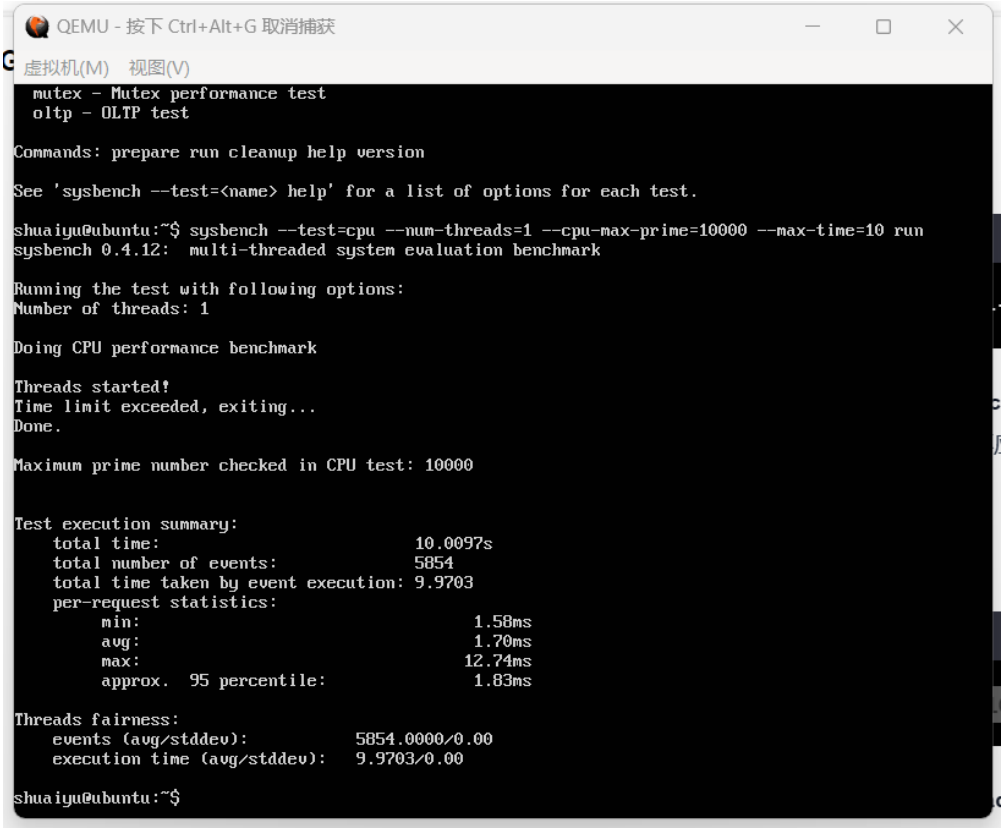
Experiment using sysbench

Environment 1:

1. CPU test:

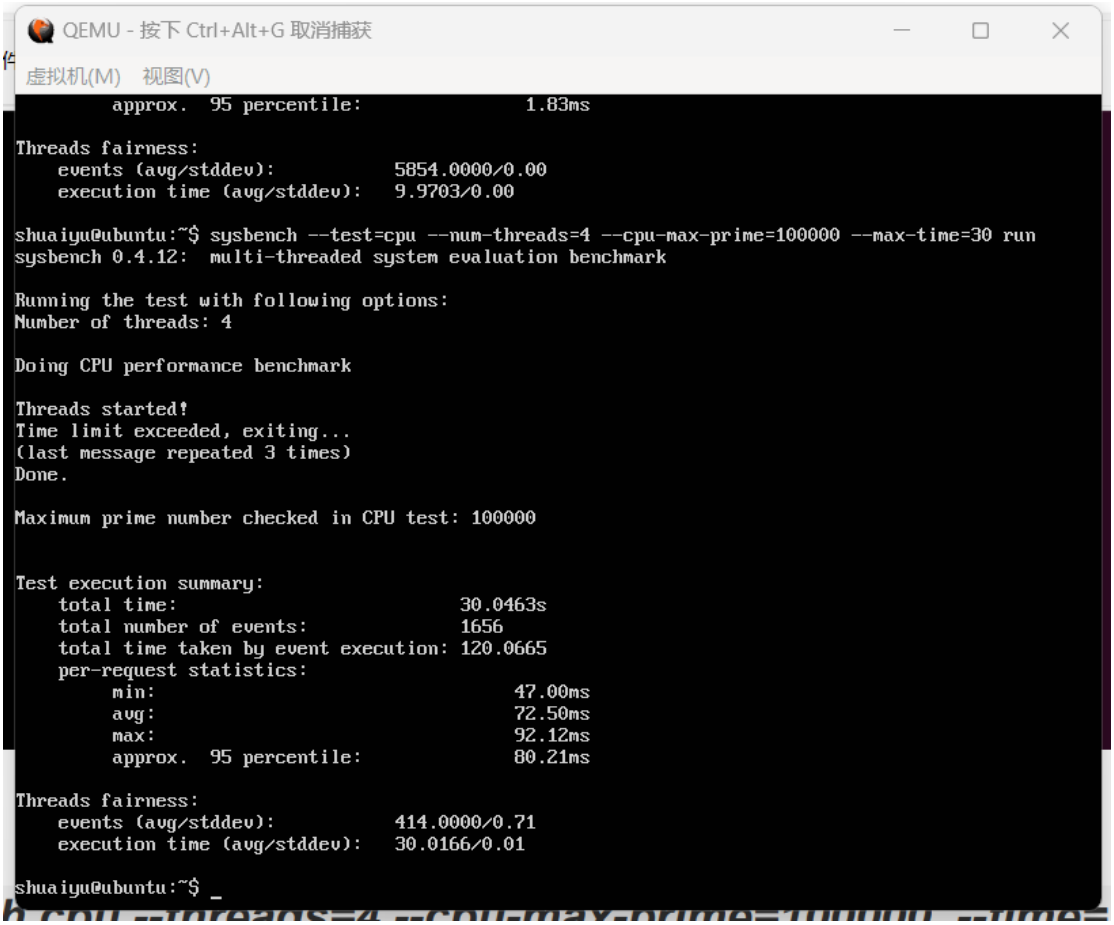
   a. Test 1:

   sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



b. Test 2:

sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



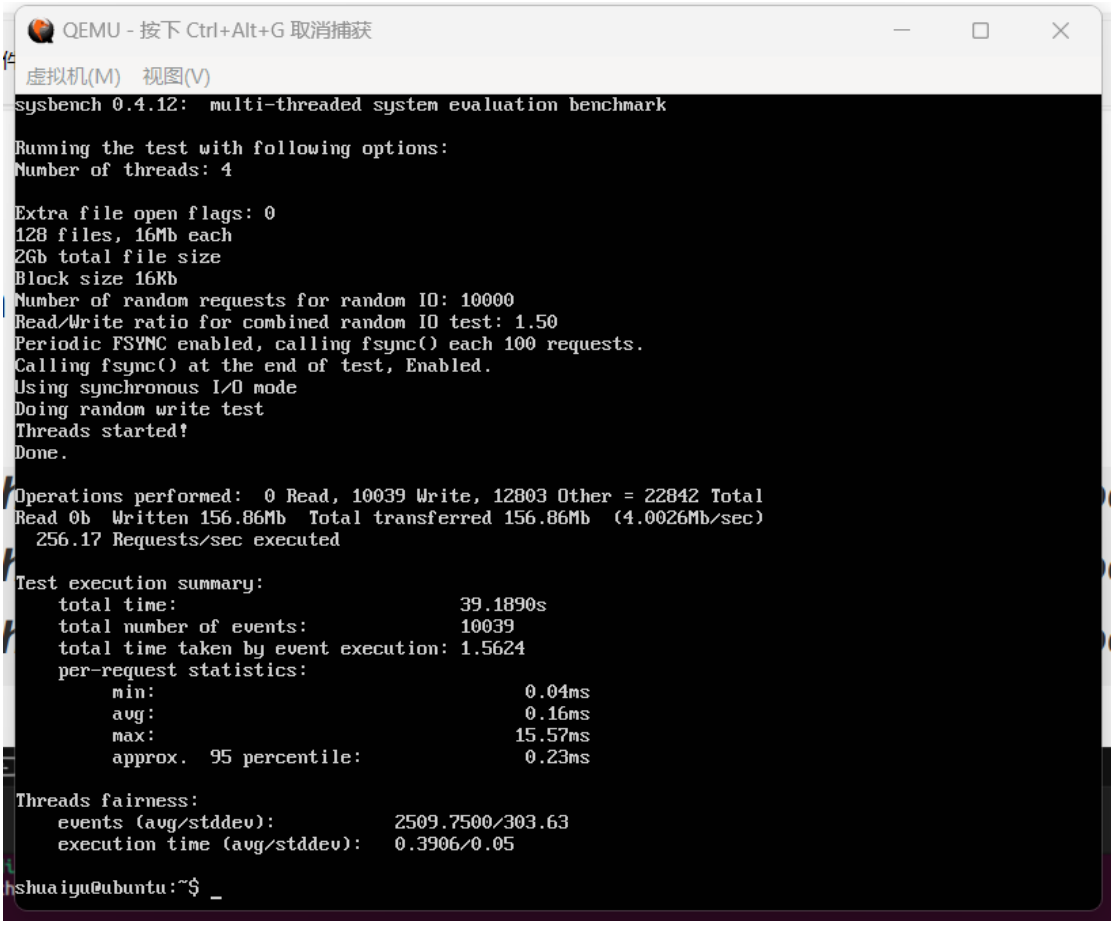
2. FILE I/O test:

a. Test 1:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup

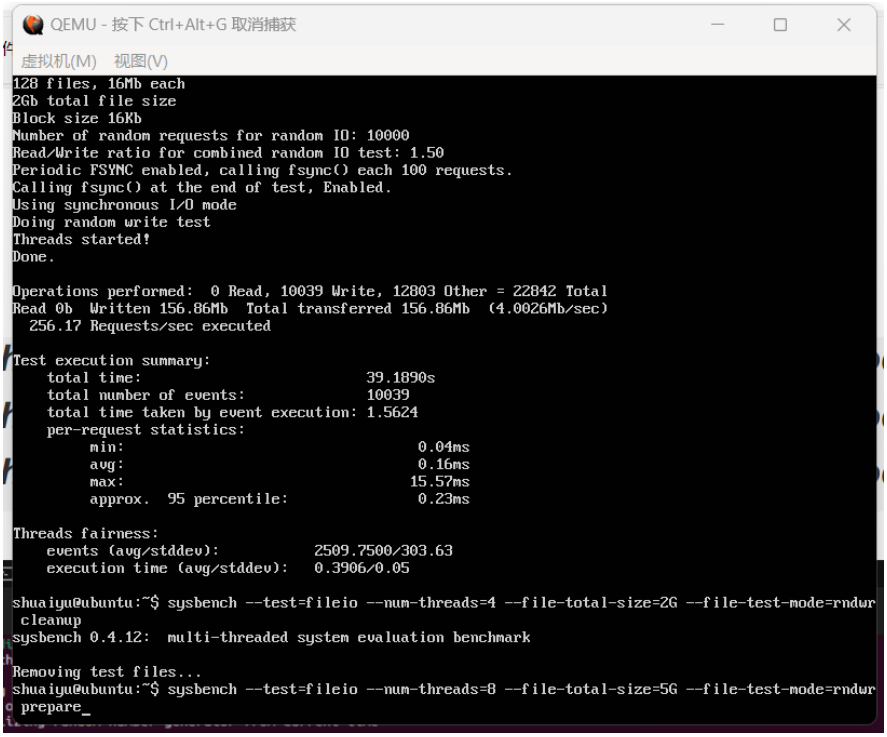


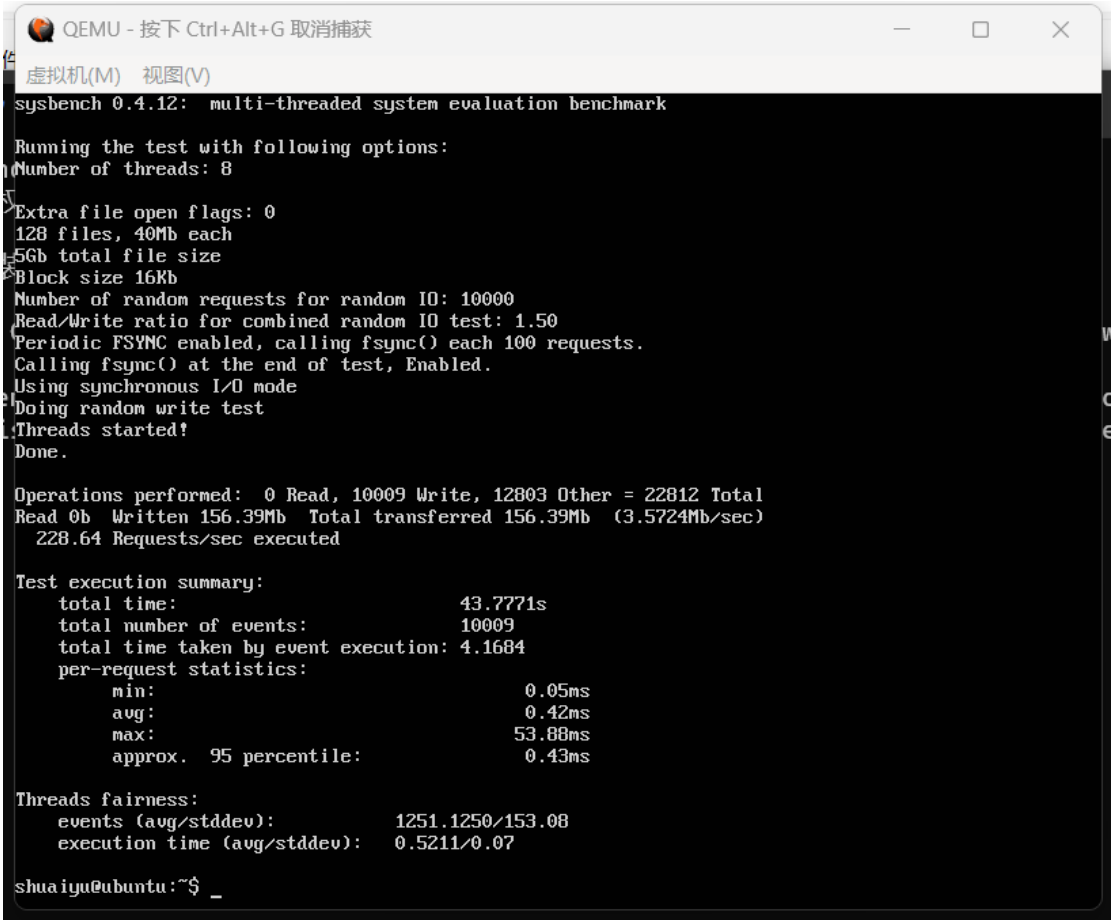
b. Test 2:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup

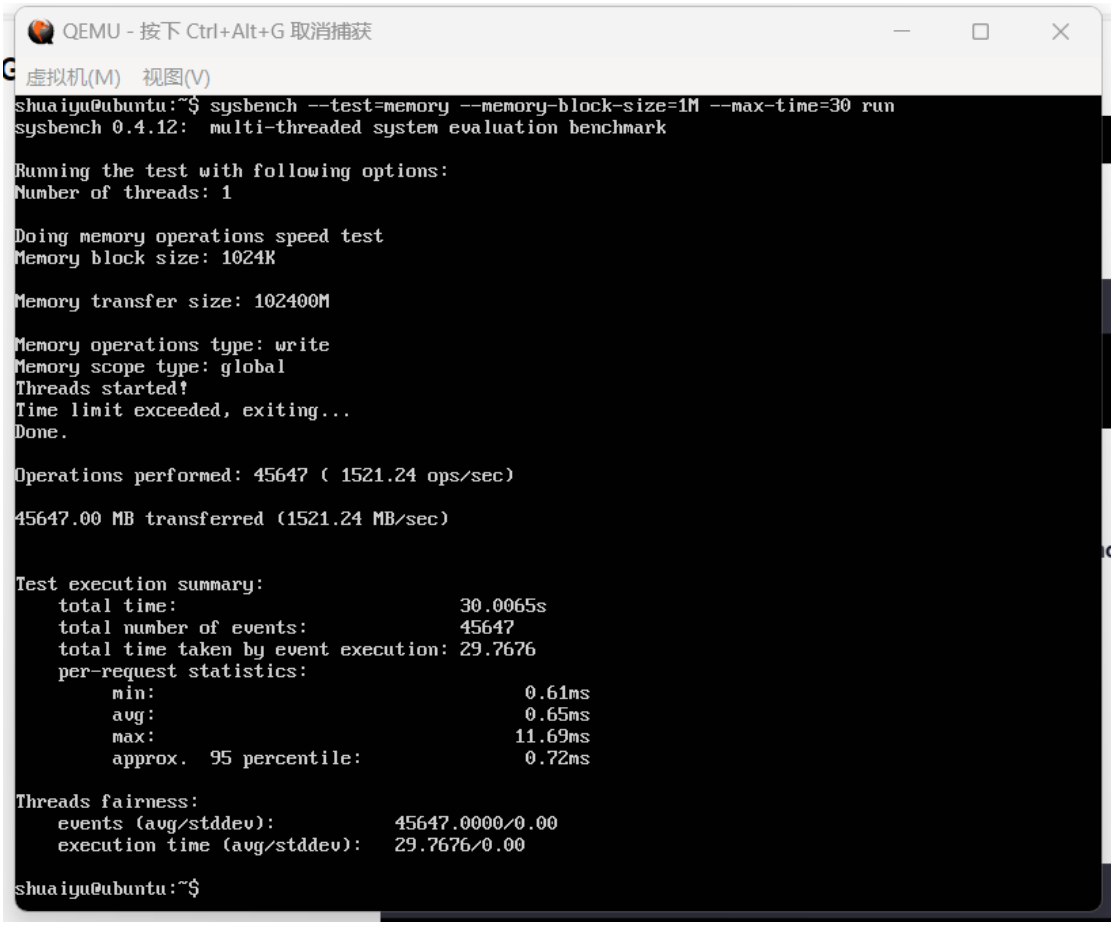




3. Memory test:

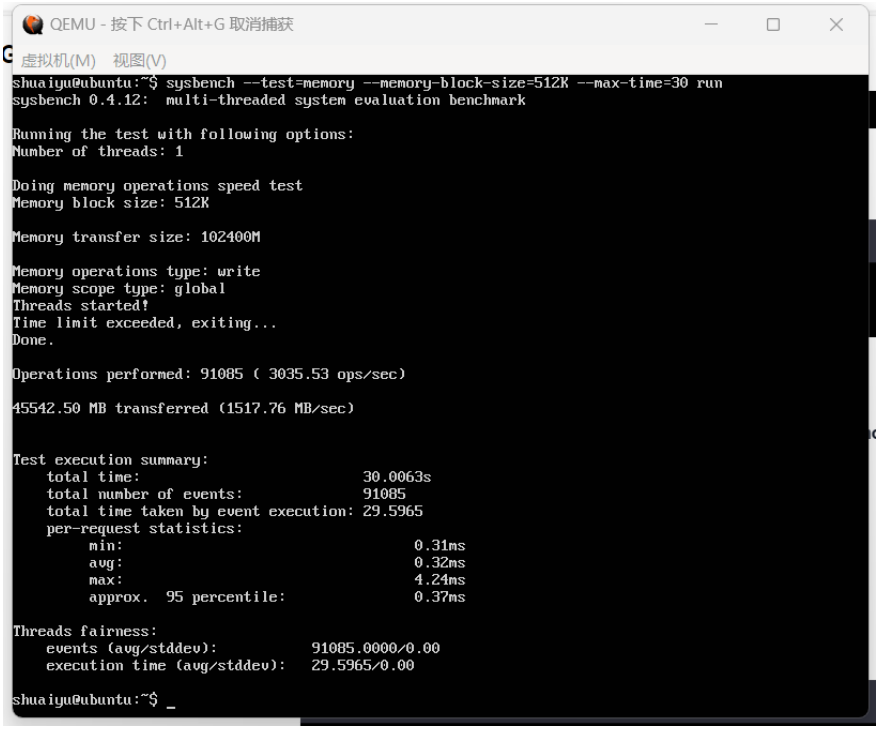
   a. Test 1:

   sysbench memory --memory-block-size=1M --time=30 run



b. Test 2:

sysbench memory --memory-block-size=512K --time=30 run

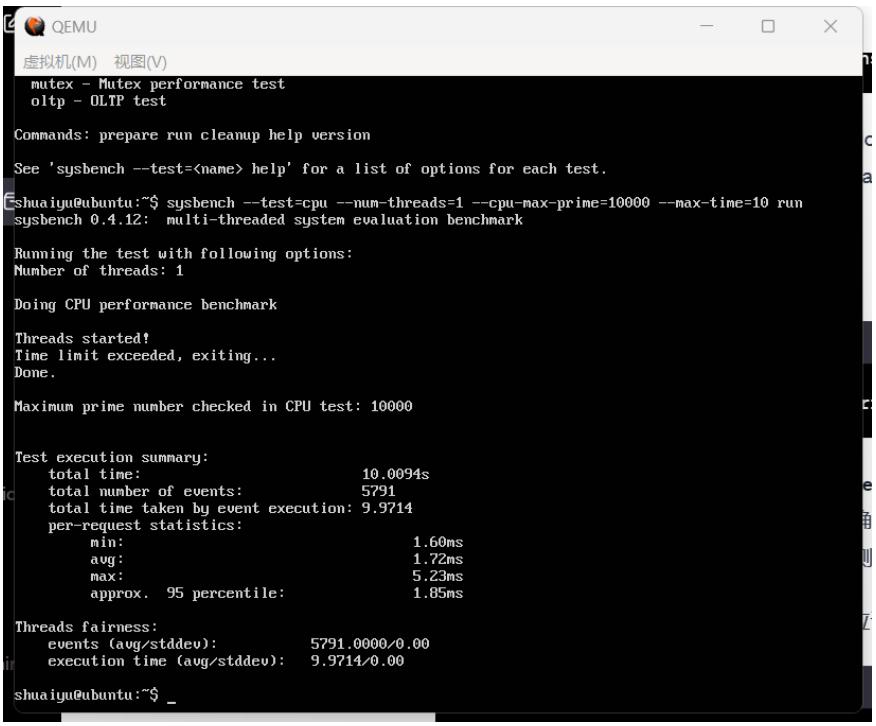


Environment 2:

1. CPU test:

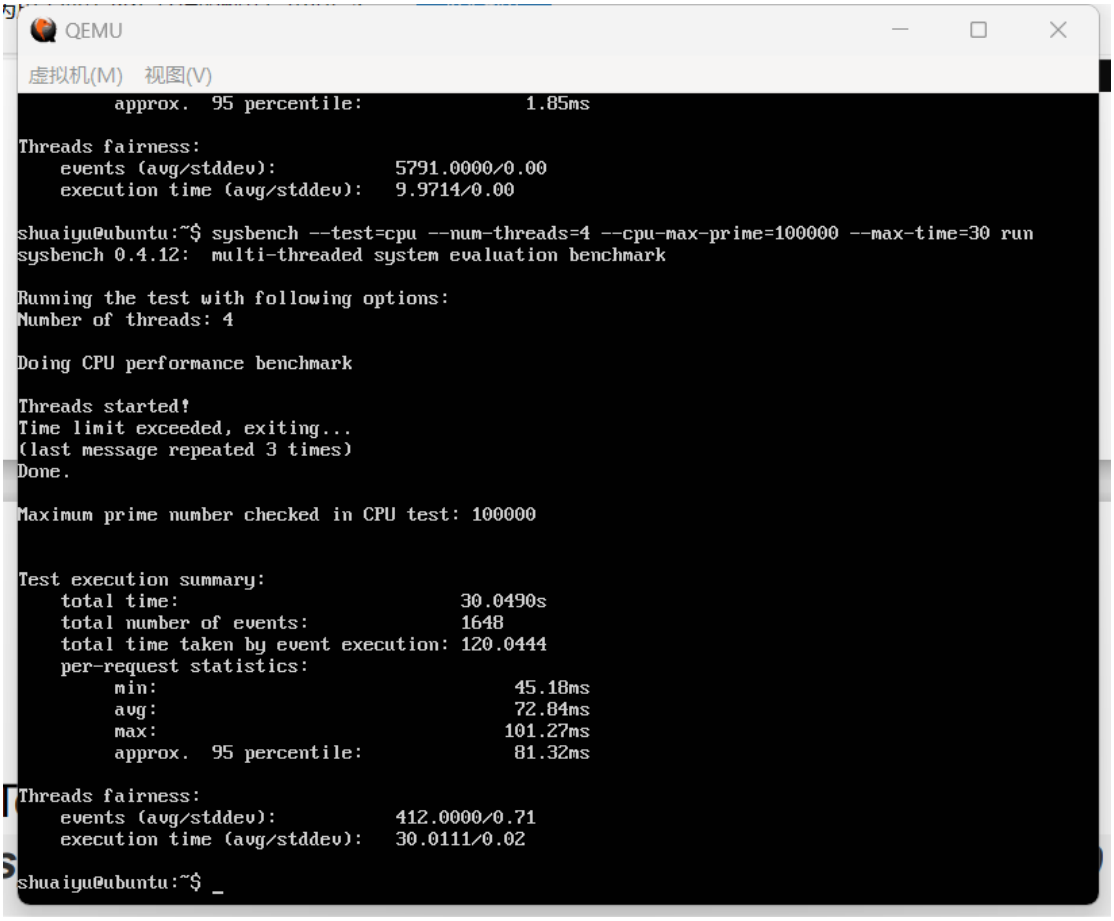
   a. Test 3:

   sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



   b. Test 4:

   sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



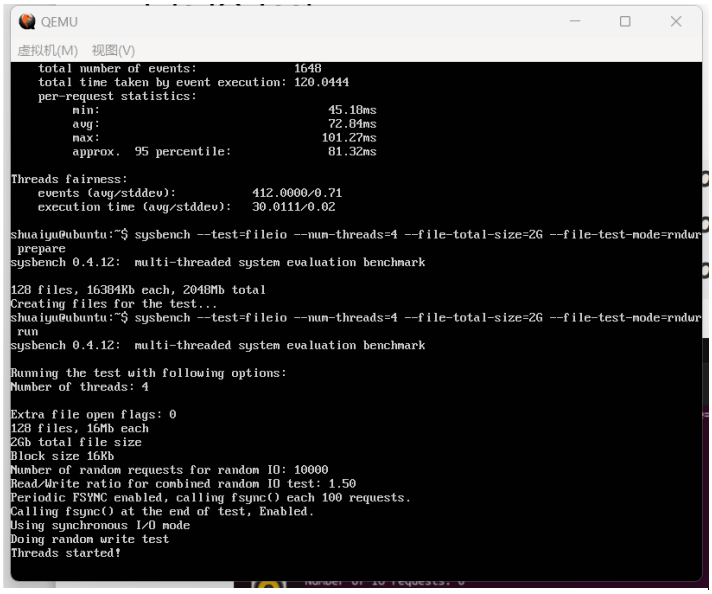
2. FILE I/O test:

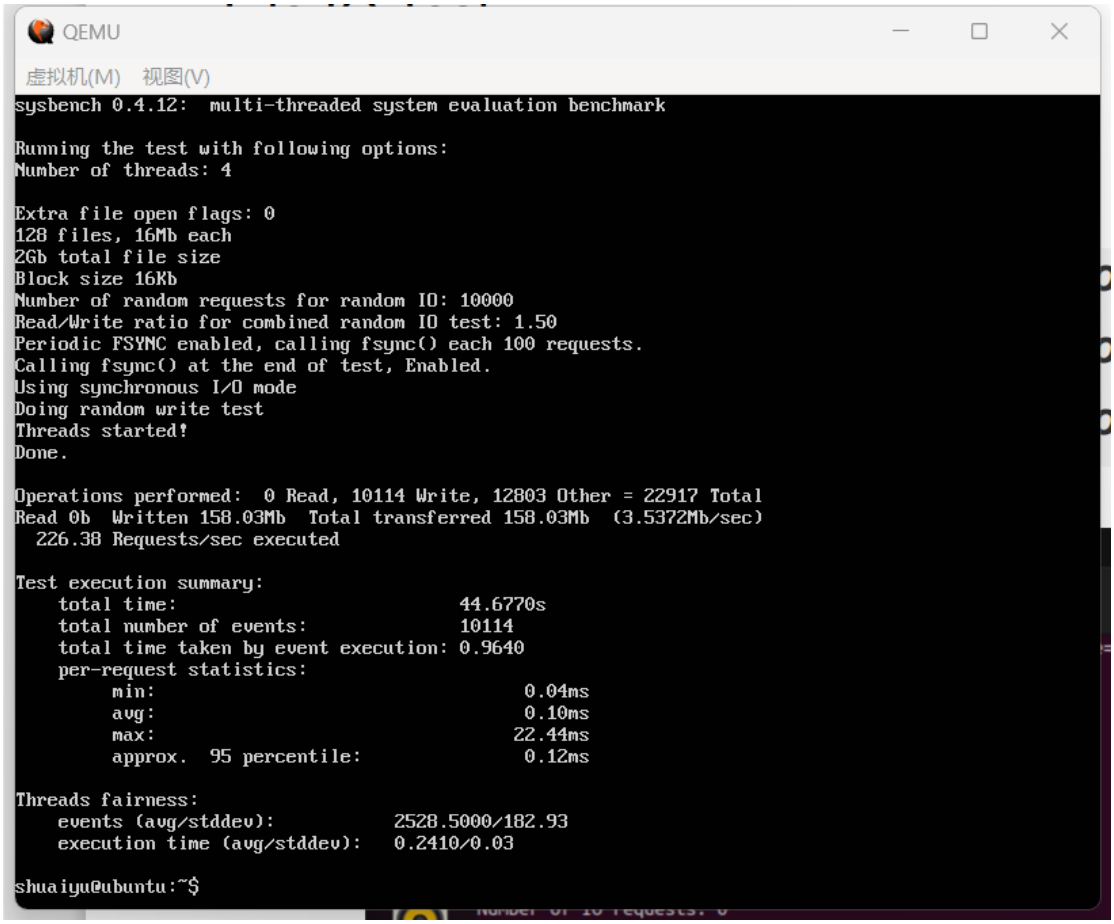
a. Test 3:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



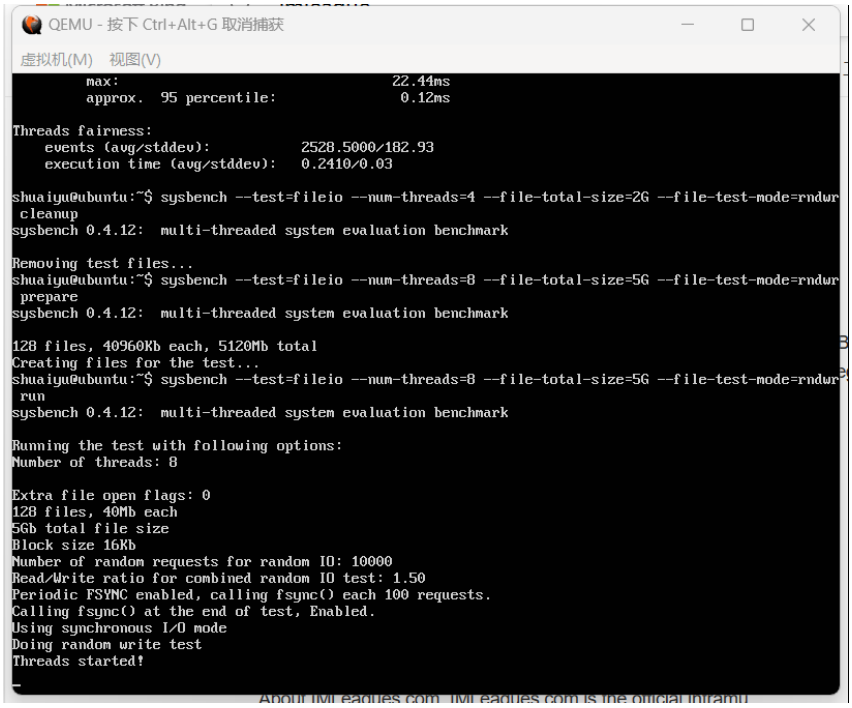


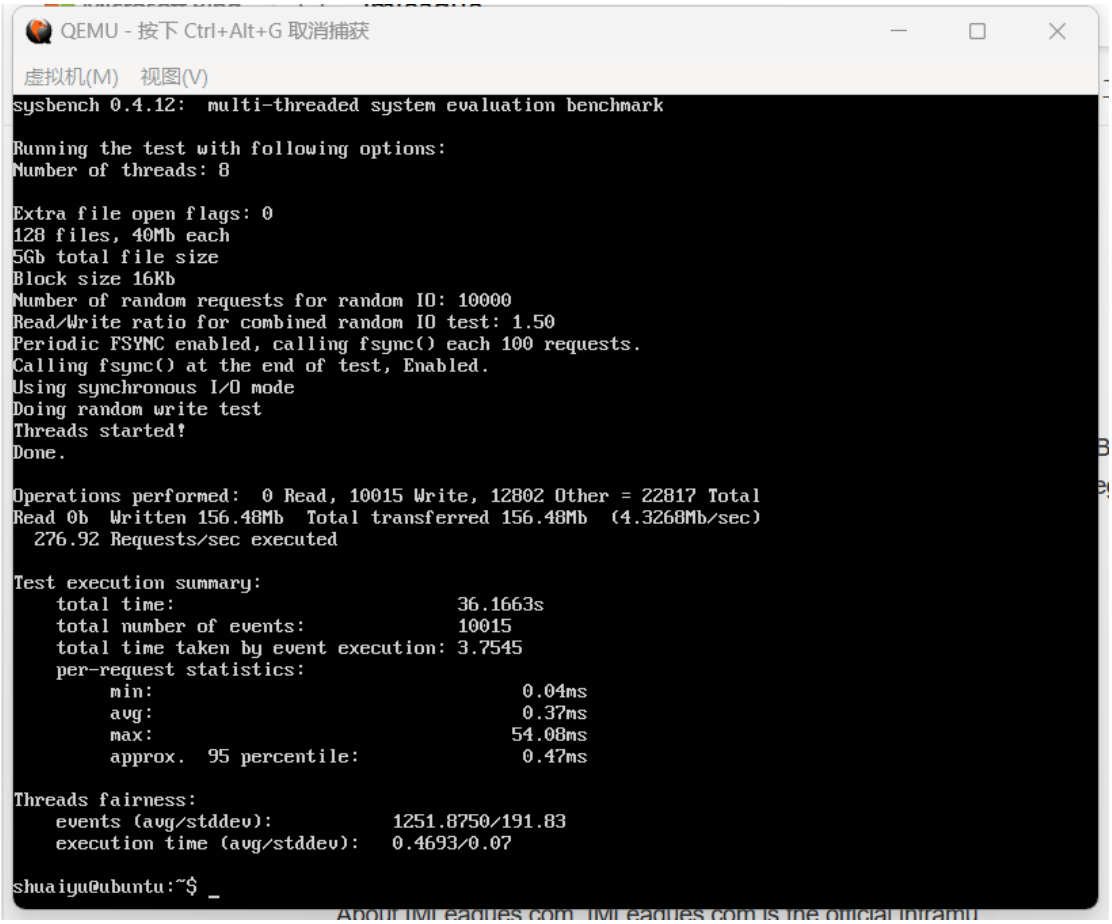
b. Test 4:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup

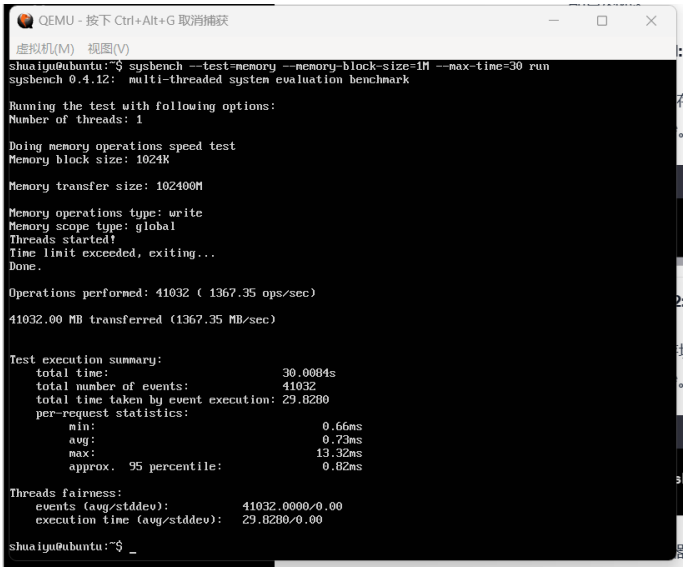




3. Memory test:

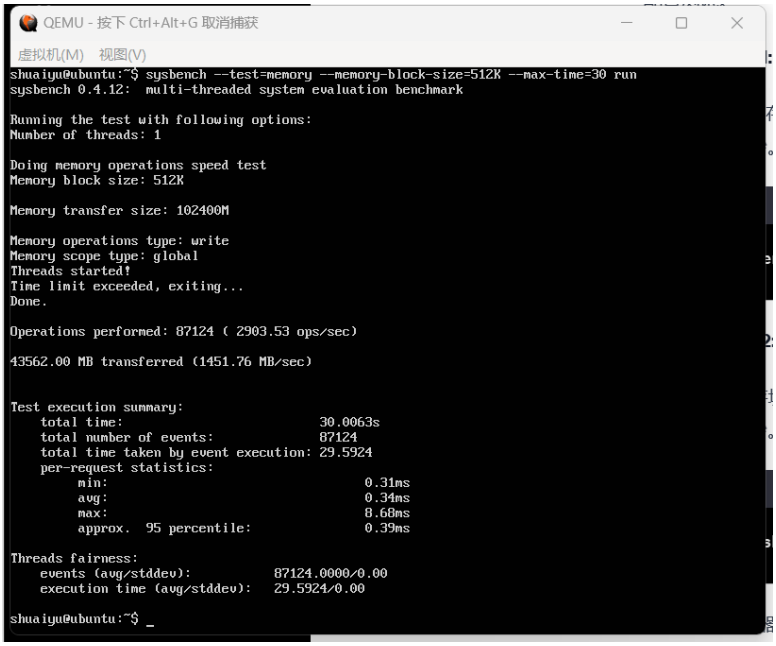
a. Test 3:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 4:

sysbench memory --memory-block-size=512K --time=30 run

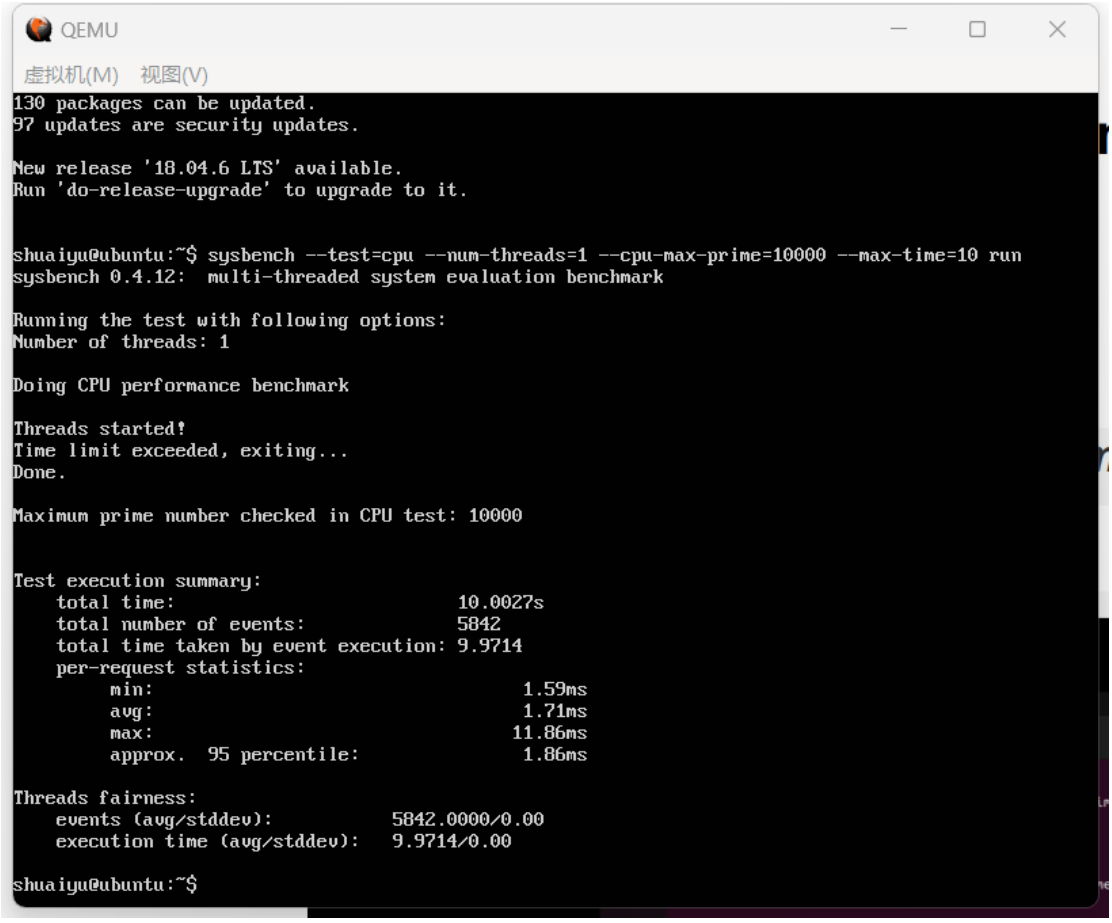


Environment 3:

1. CPU test:

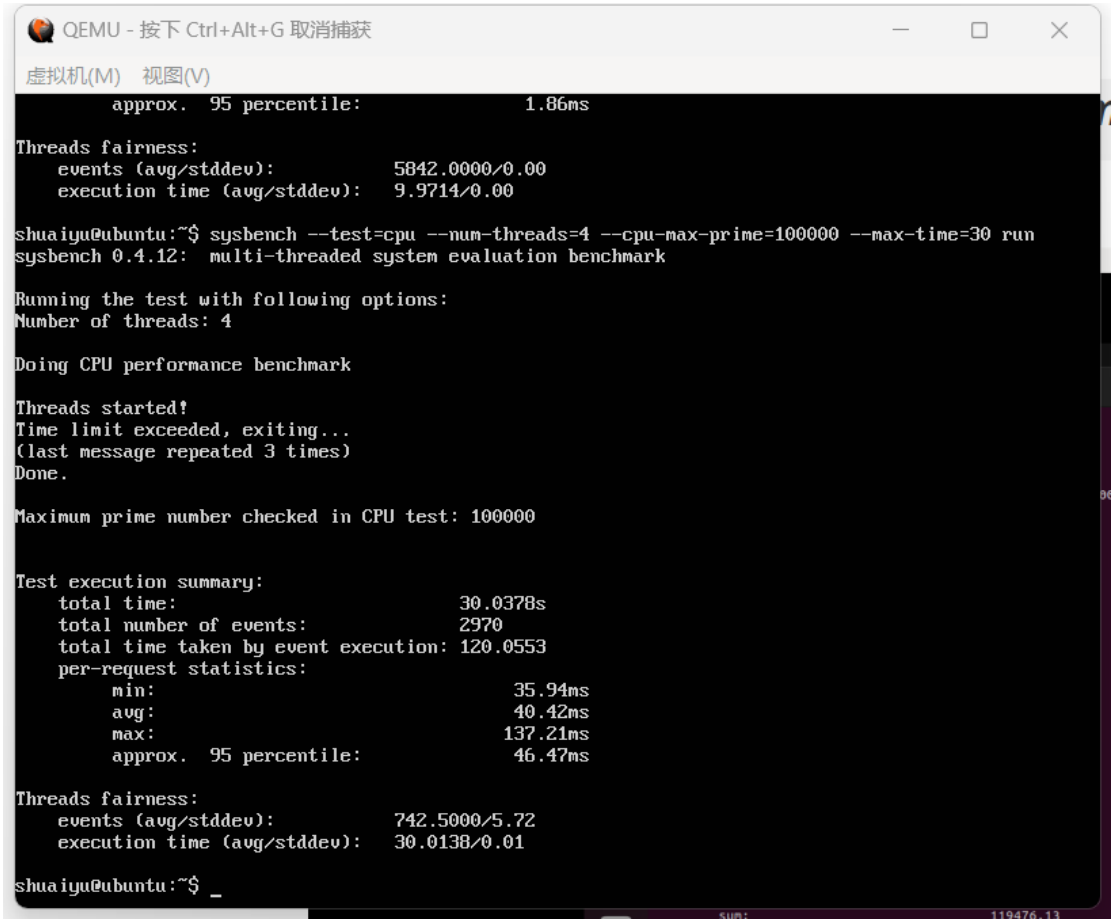
   a. Test 5:

   sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



   b. Test 6:

sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



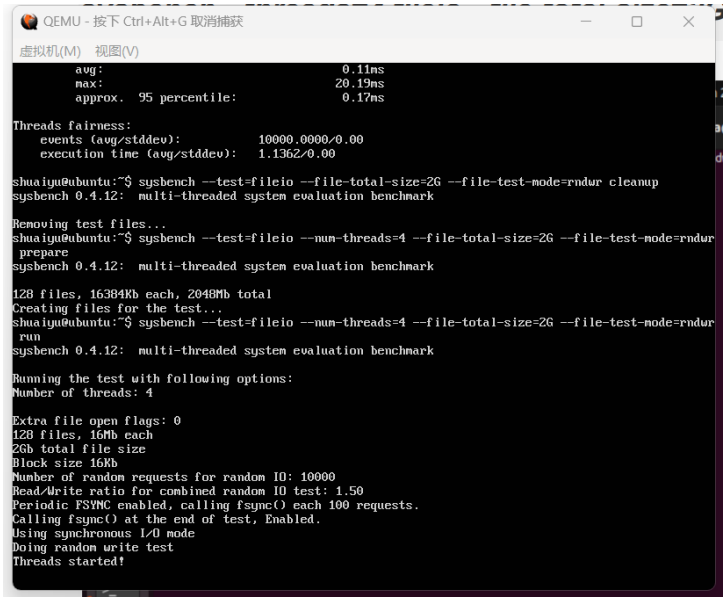
2. FILE I/O test:

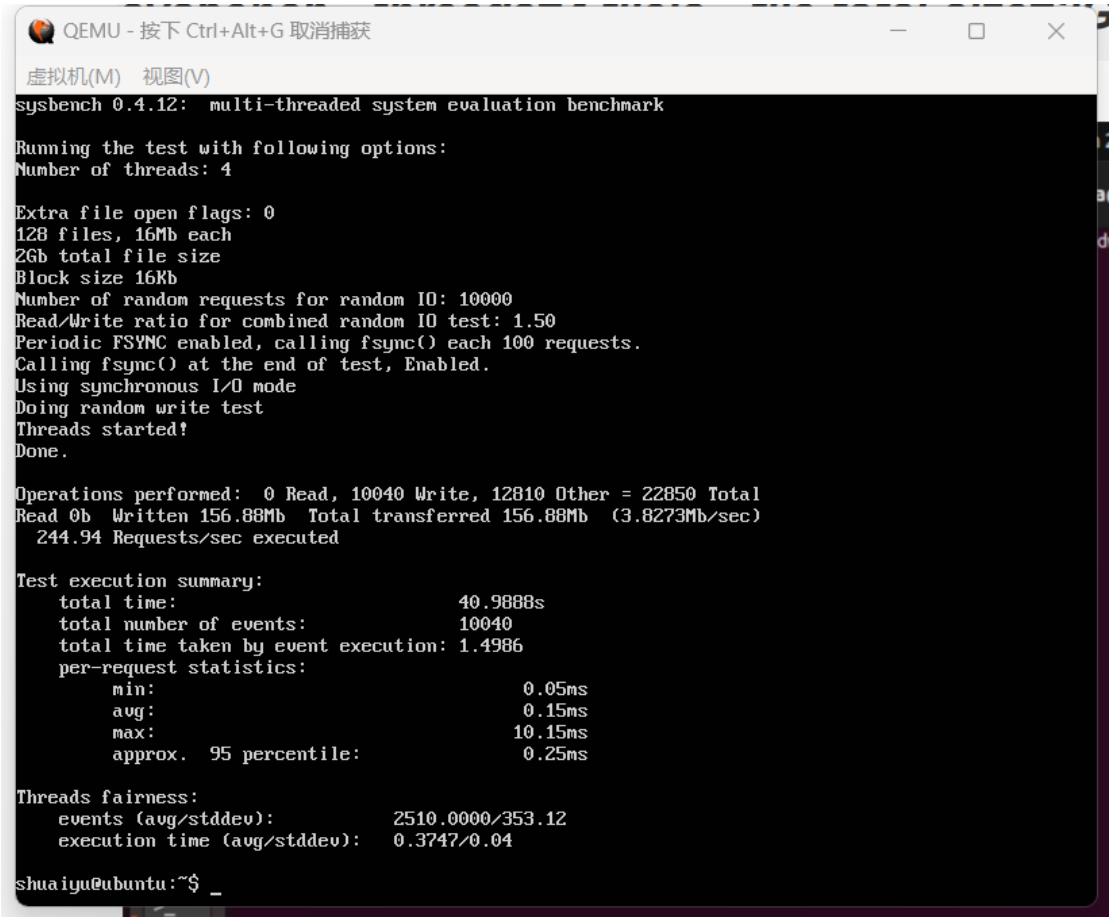
a. Test 5:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



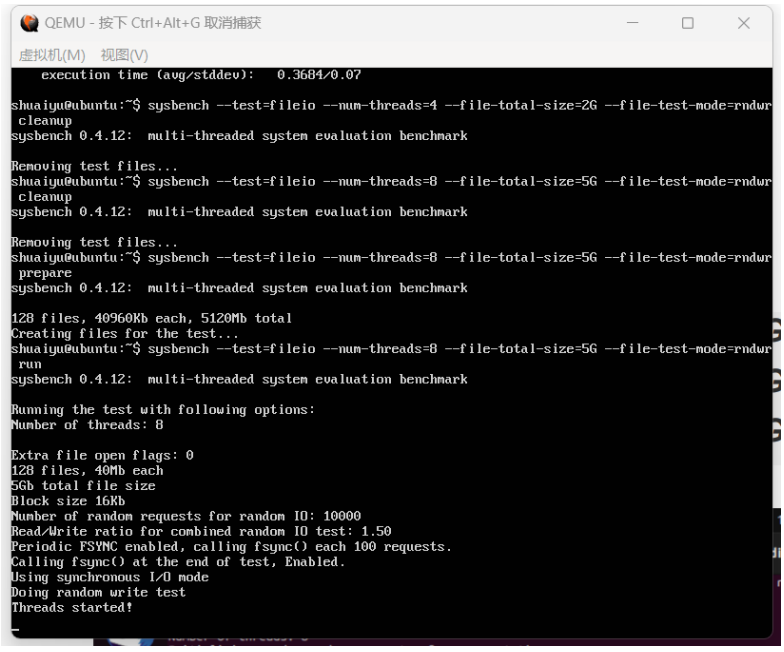


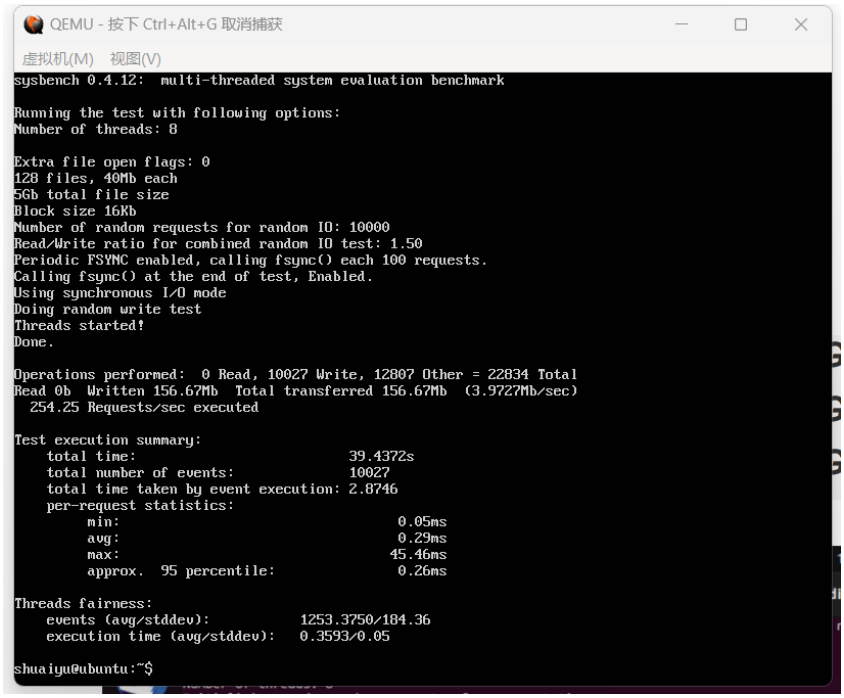
b. Test 6:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup

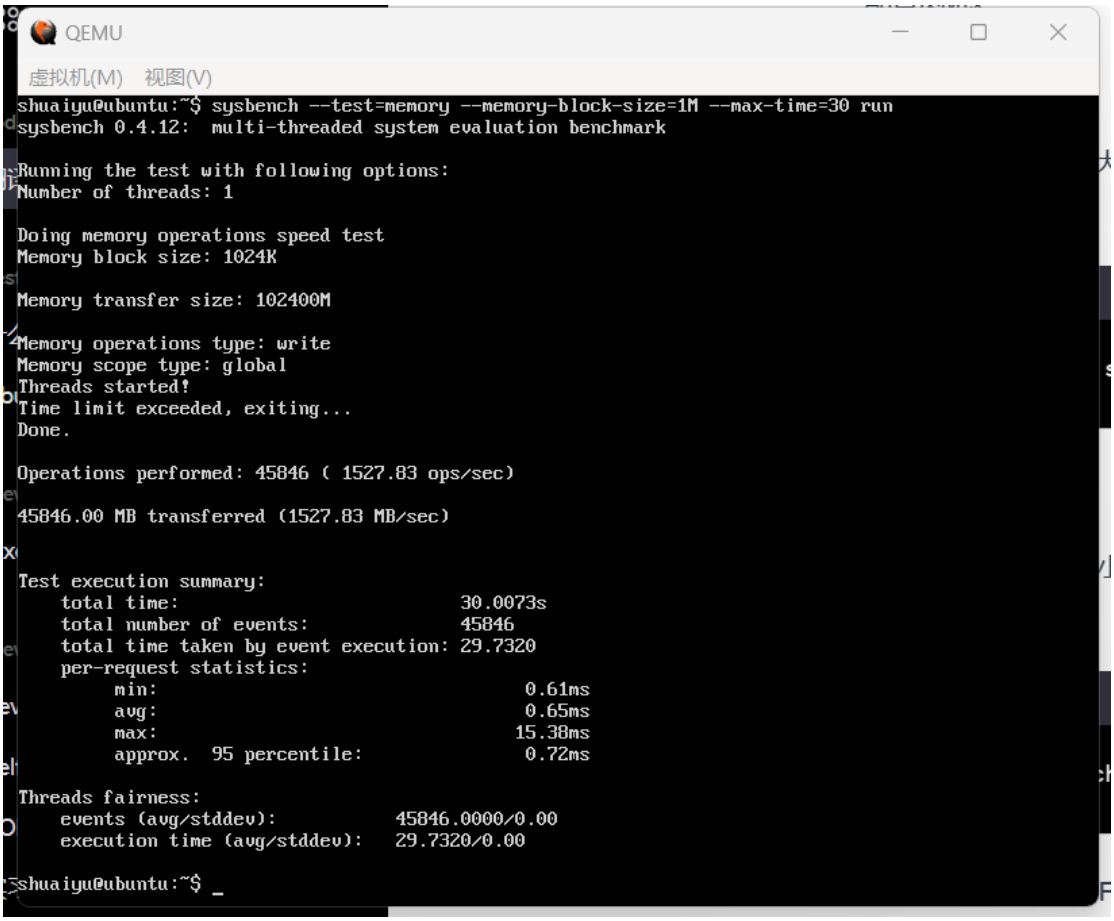




3. Memory test:

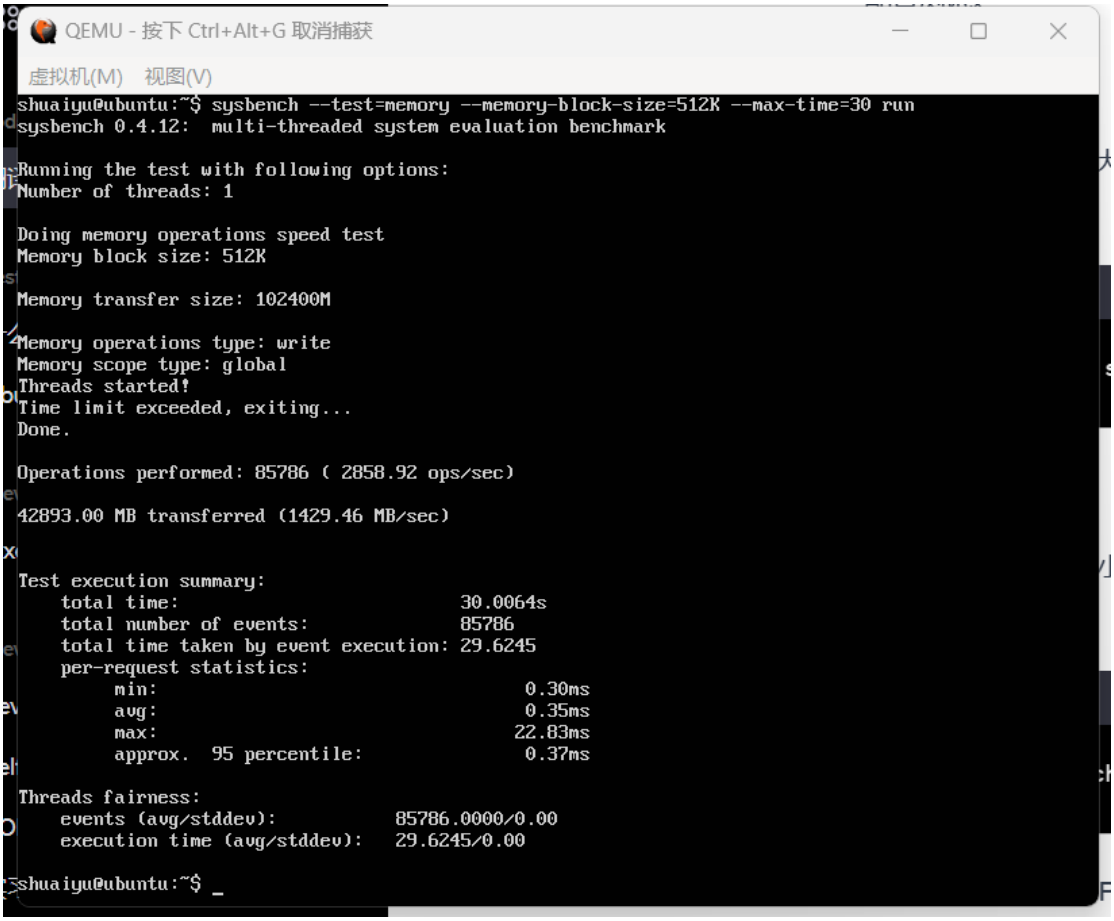
a. Test 5:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 6:

sysbench memory --memory-block-size=512K --time=30 run

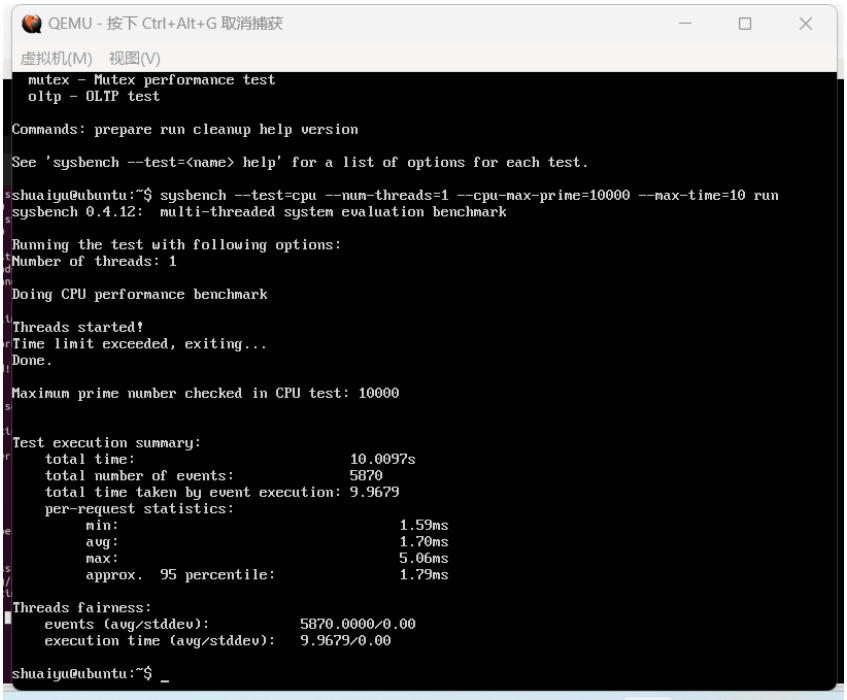


Environment 4:

1. CPU test:

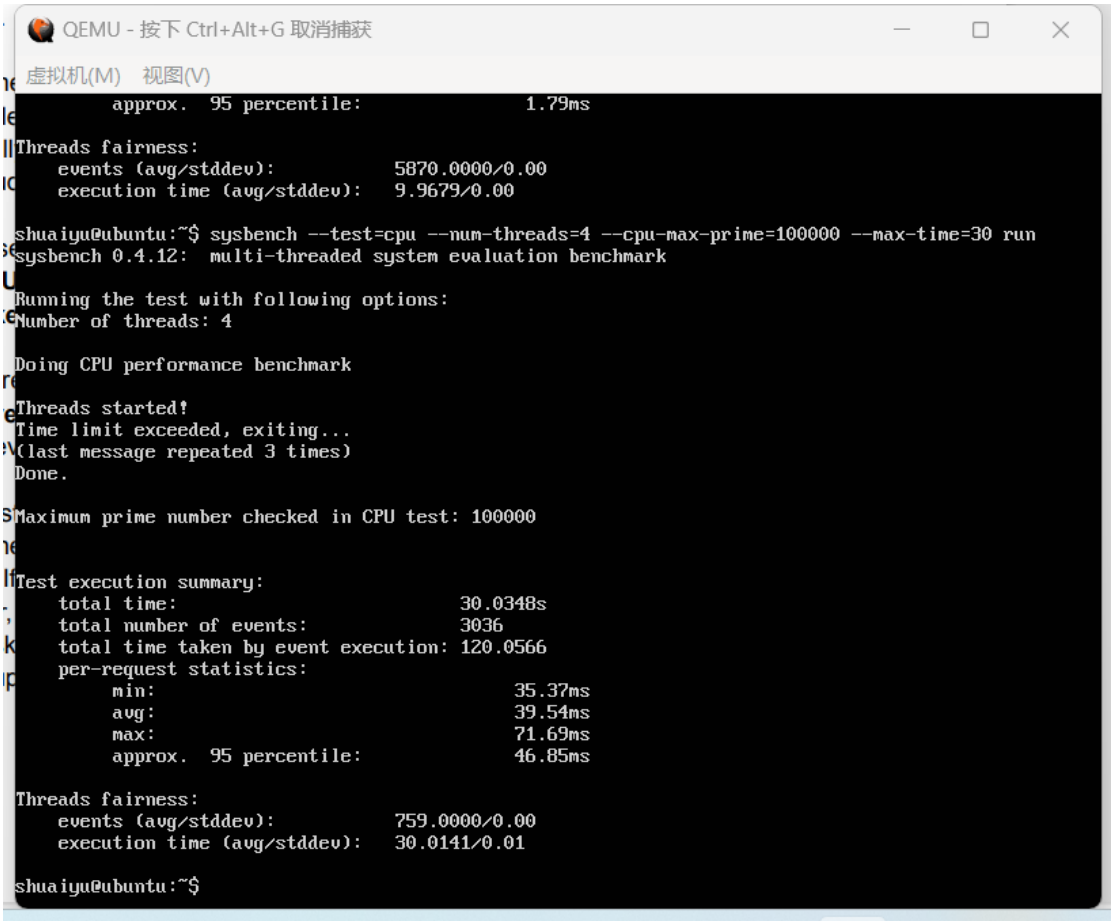
   a. Test 7:

   sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



b. Test 8:

sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



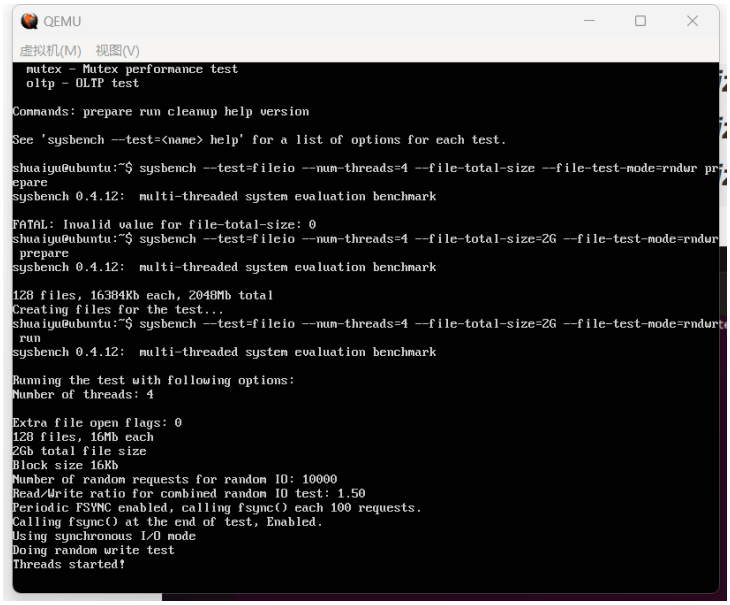
2. FILE I/O test:

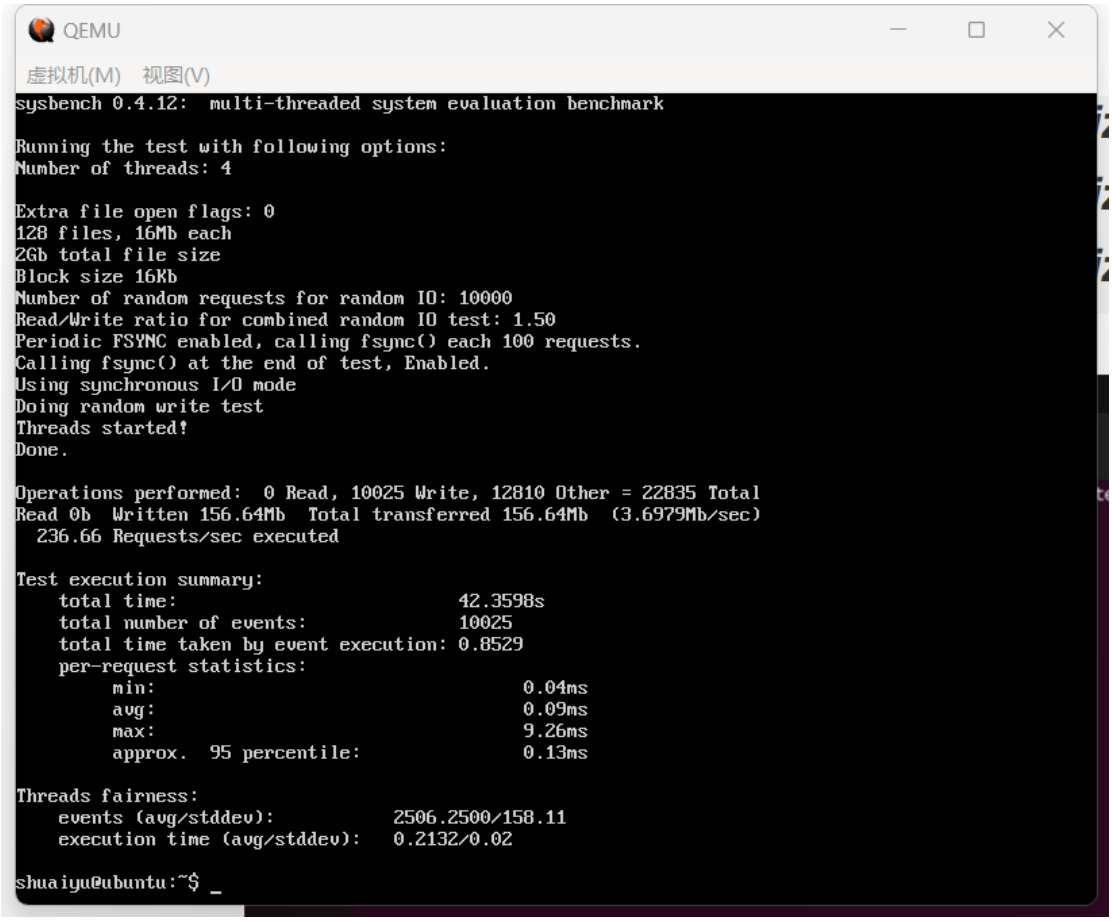
a. Test 7:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



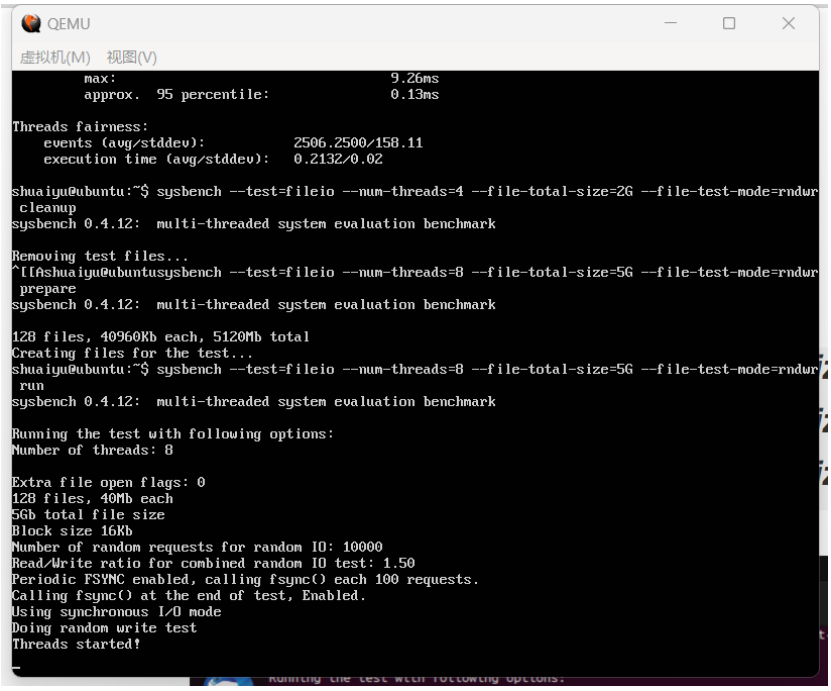


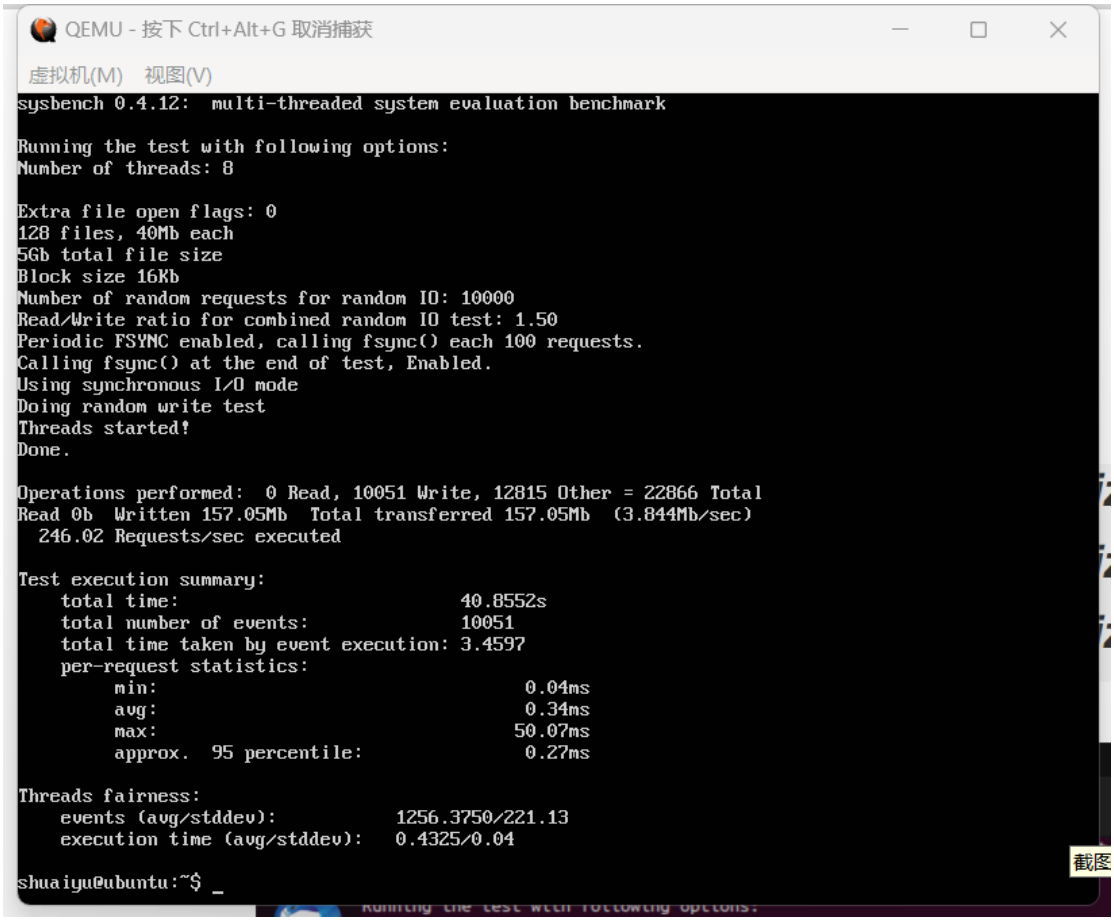
b. Test 8:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup

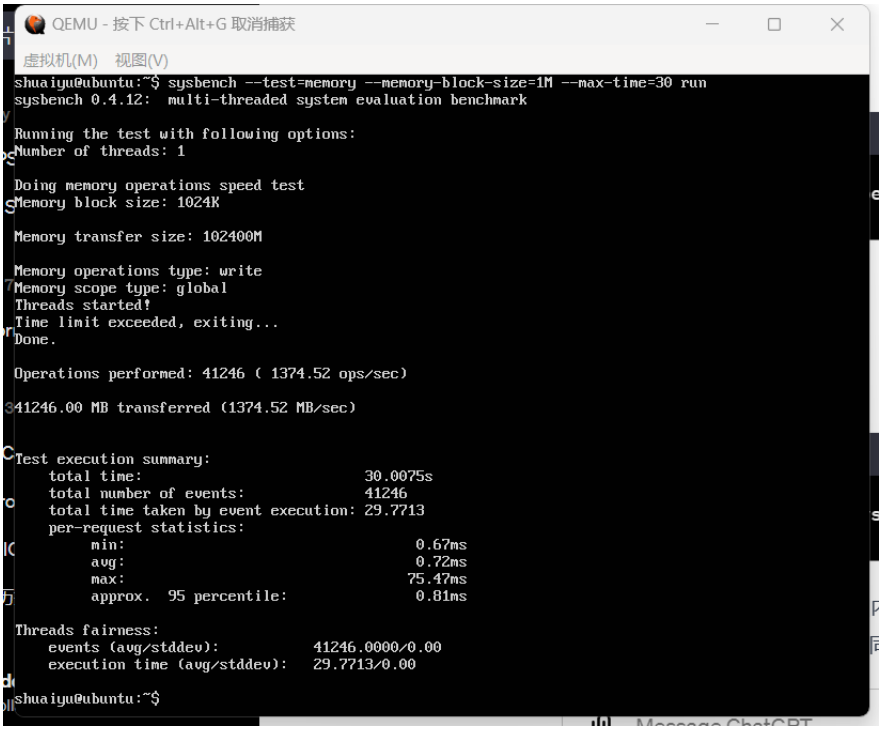




3. Memory test:

a. Test 7:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 8:

sysbench memory --memory-block-size=512K --time=30 run

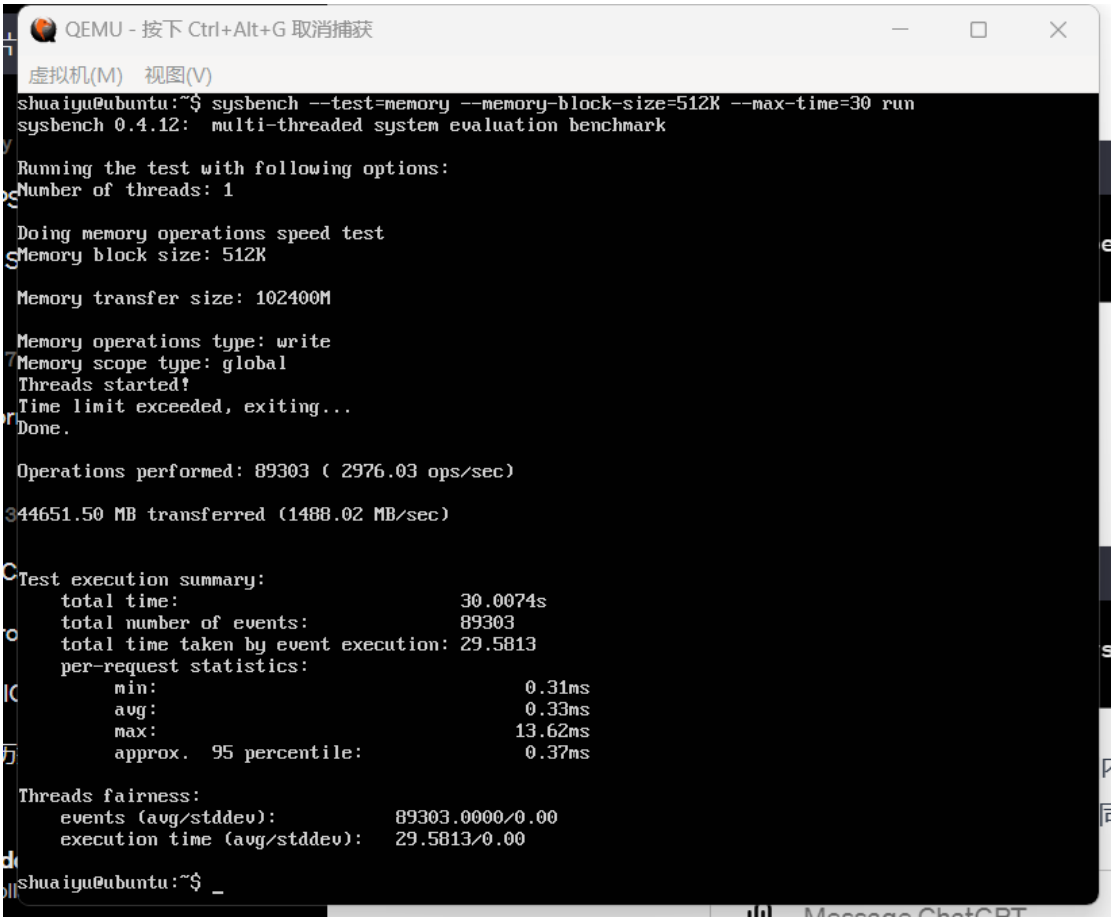
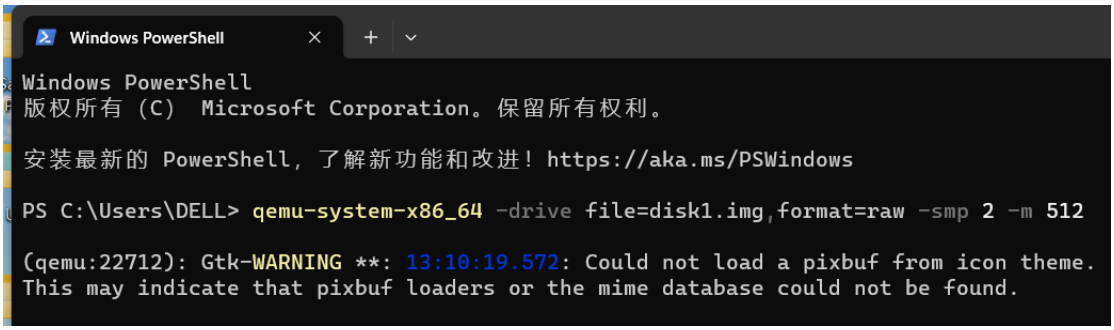
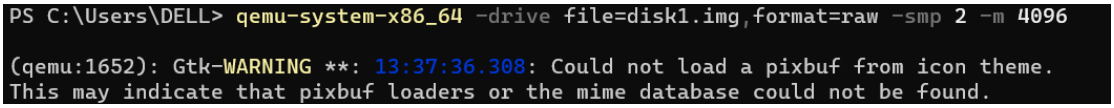


Image 2: disk.img format=raw

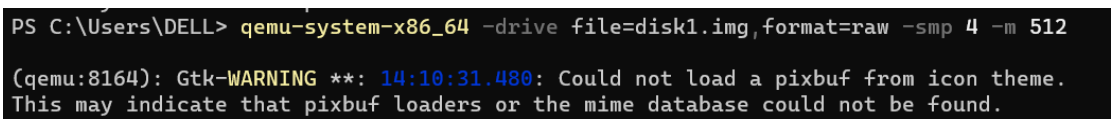
Environment 1:



Environment 2:



Environment 3:



Environment 4:

```
PS C:\Users\DELL> qemu-system-x86_64 -drive file=disk1.img,format=raw -smp 4 -m 4096

(qemu:5336): Gtk-WARNING **: 14:24:01.765: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
```

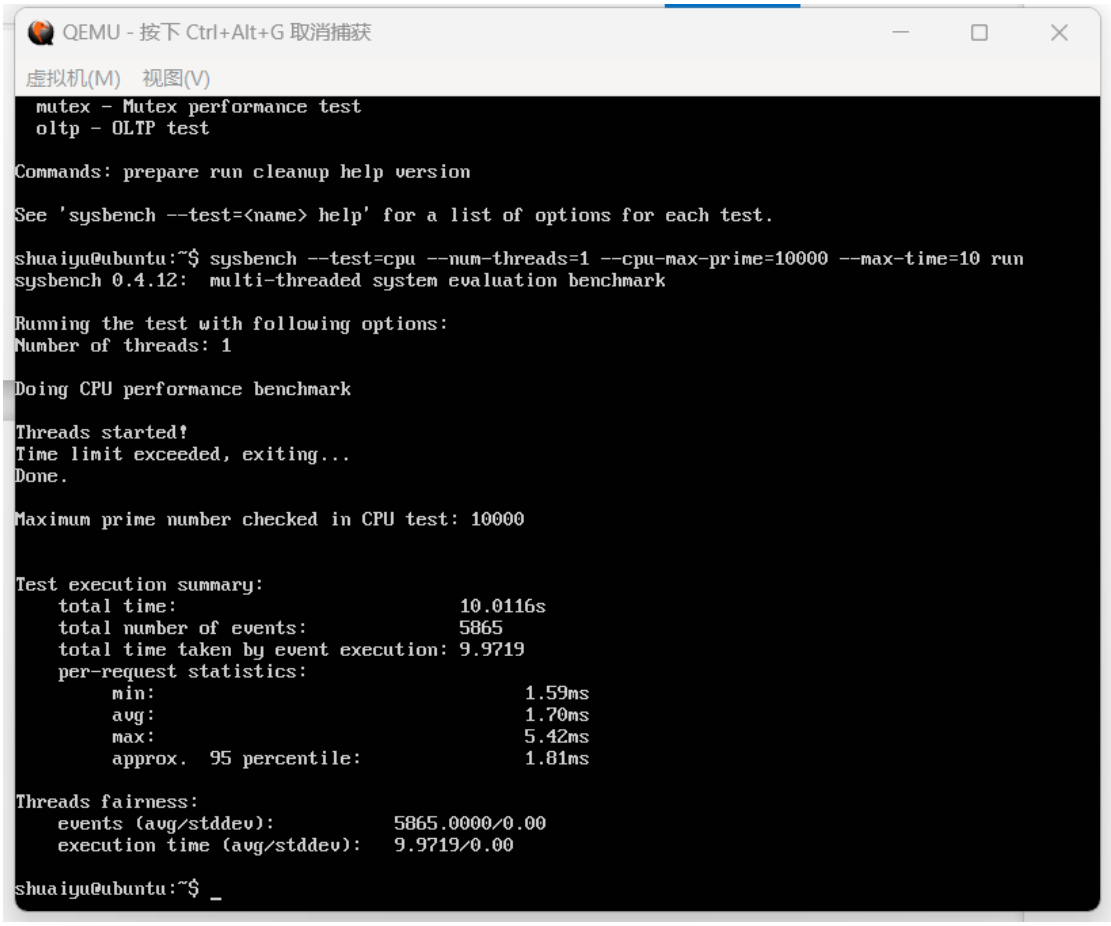
Experiment using sysbench

Environment 1:

1. CPU test:

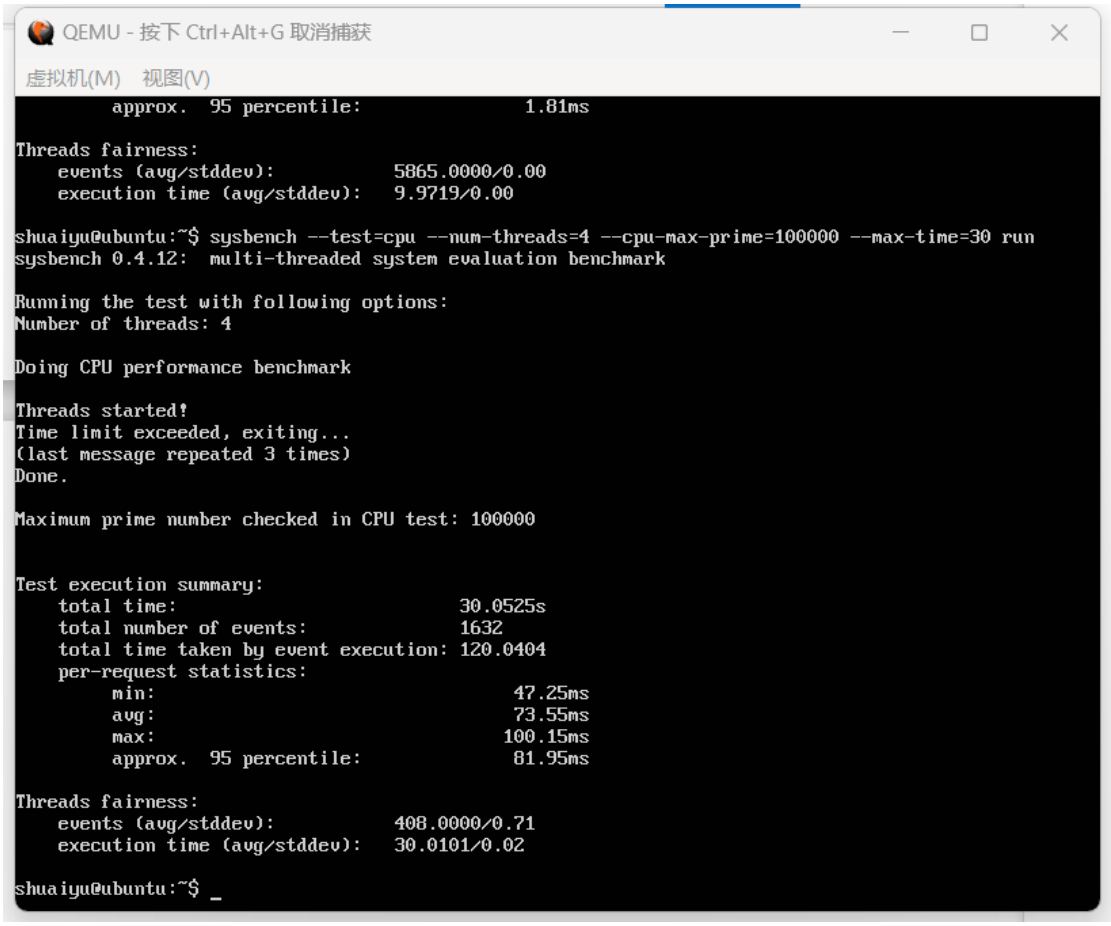
a. Test 1:

sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



b. Test 2:

sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



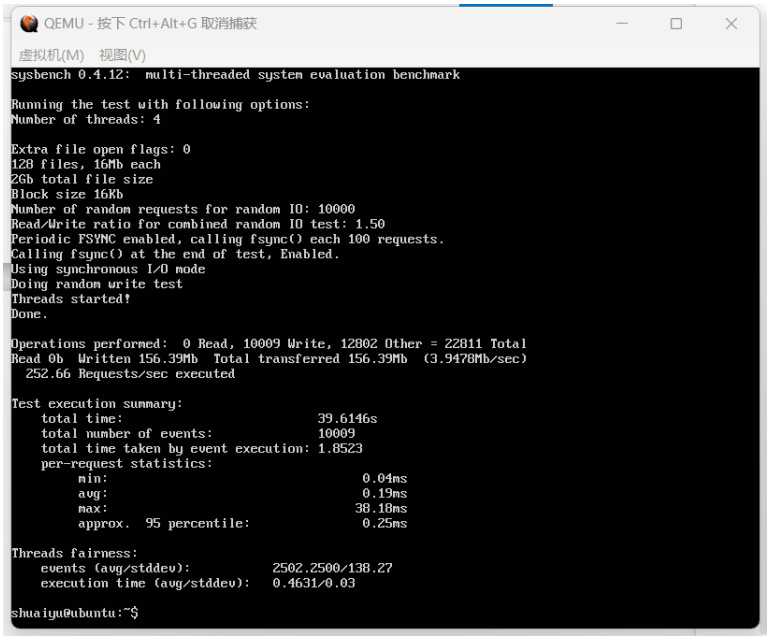
2. FILE I/O test:

a. Test 1:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup

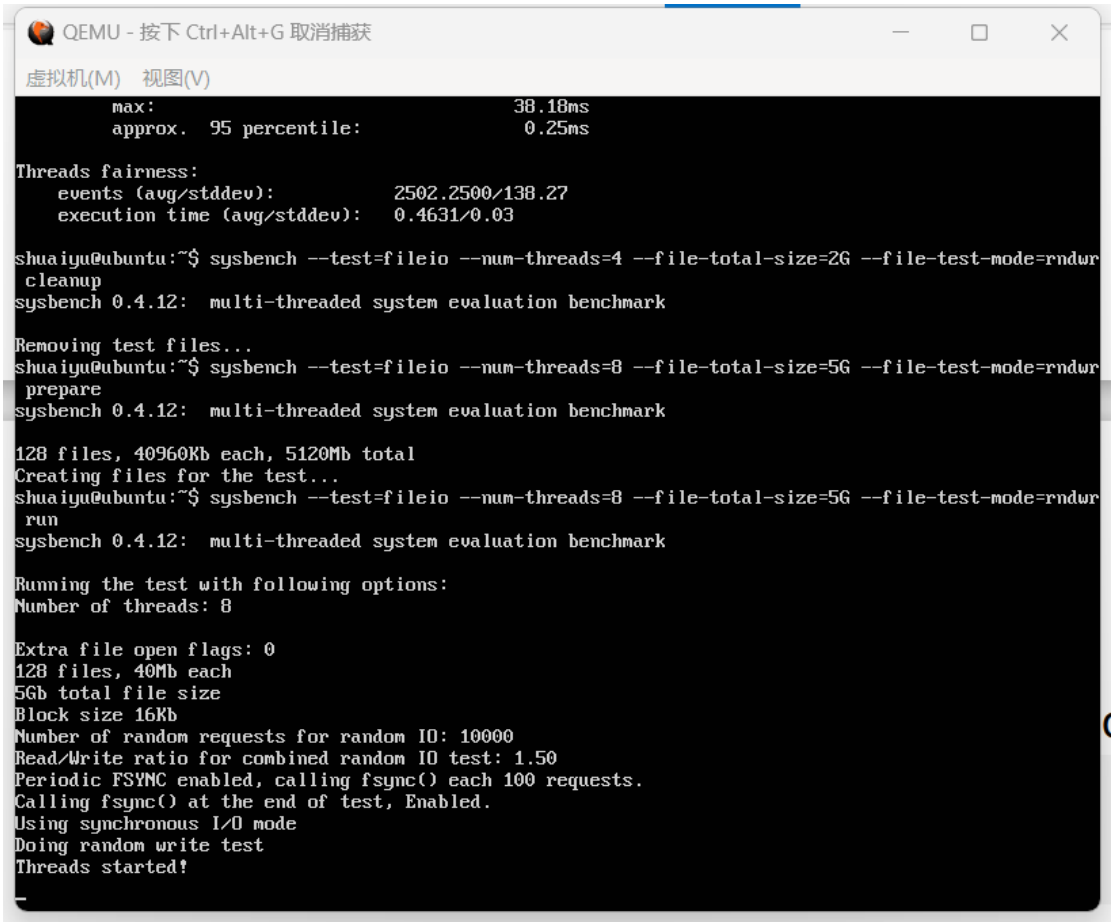


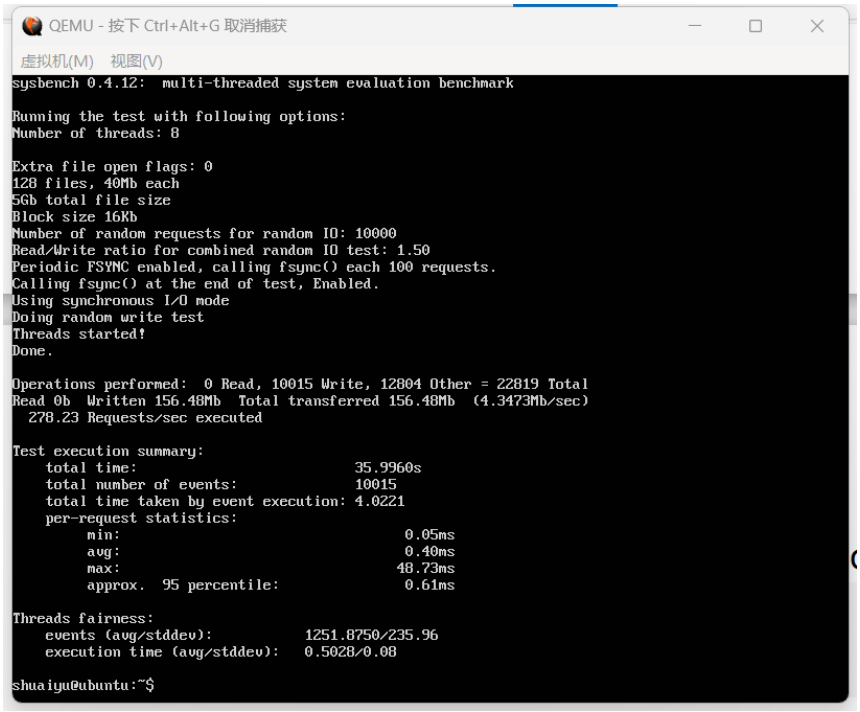
b. Test 2:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup

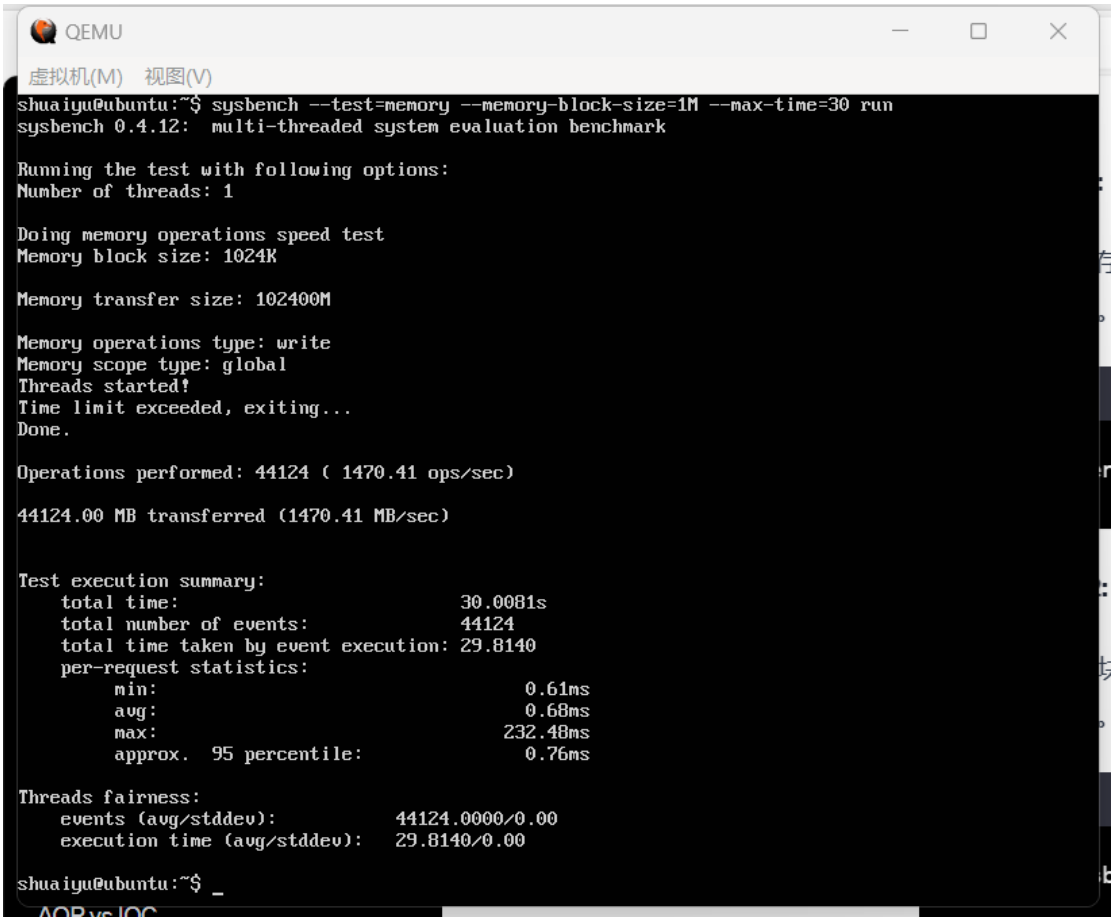




3. Memory test:

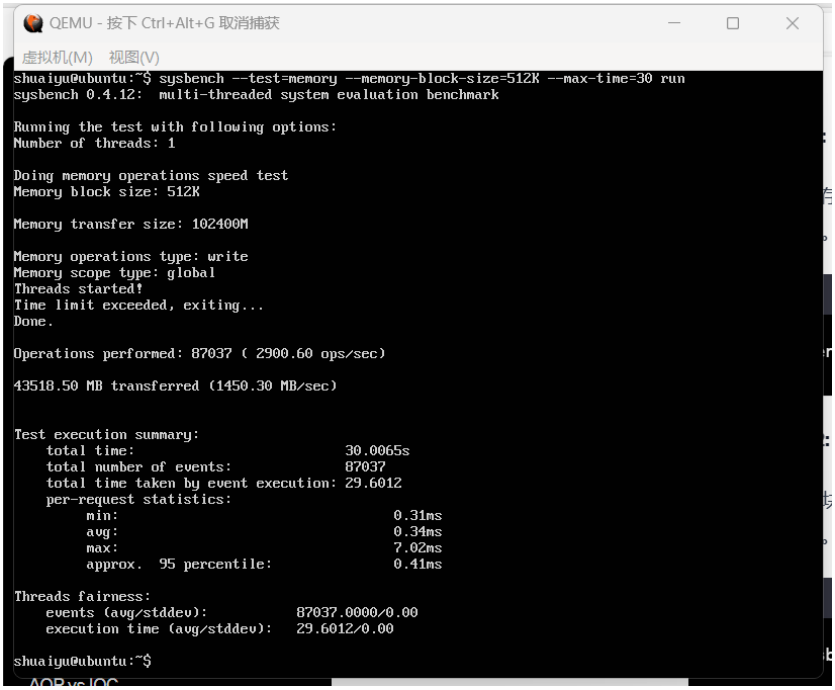
a. Test 1:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 2:

sysbench memory --memory-block-size=512K --time=30 run

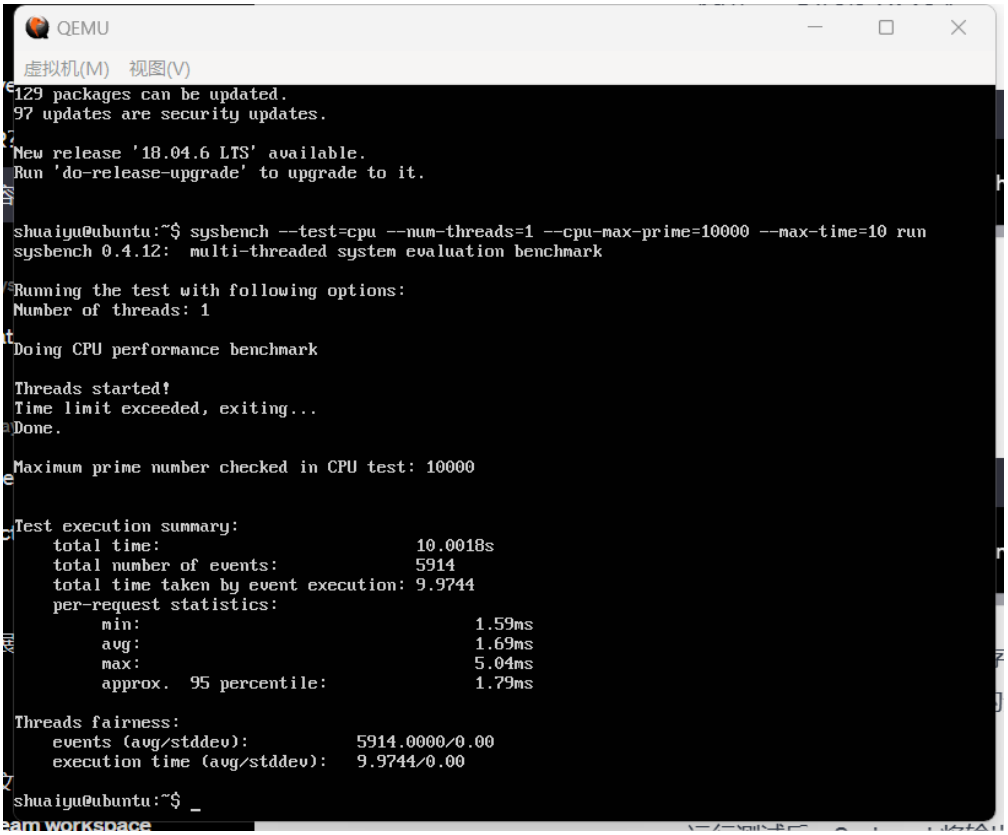


Environment 2:

1. CPU test:

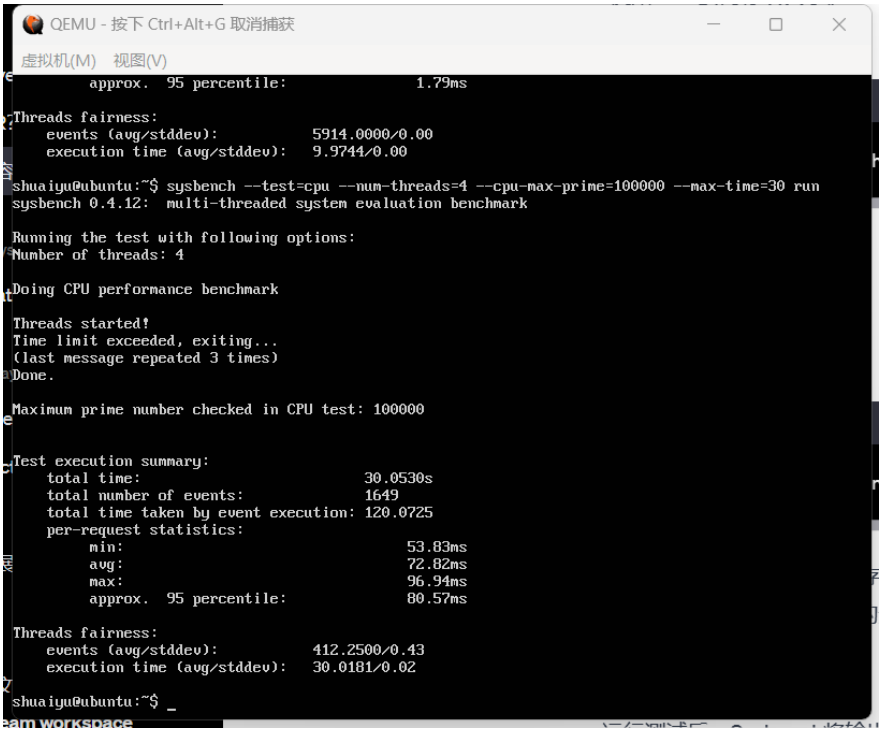
a. Test 3:

sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



b. Test 4:

sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



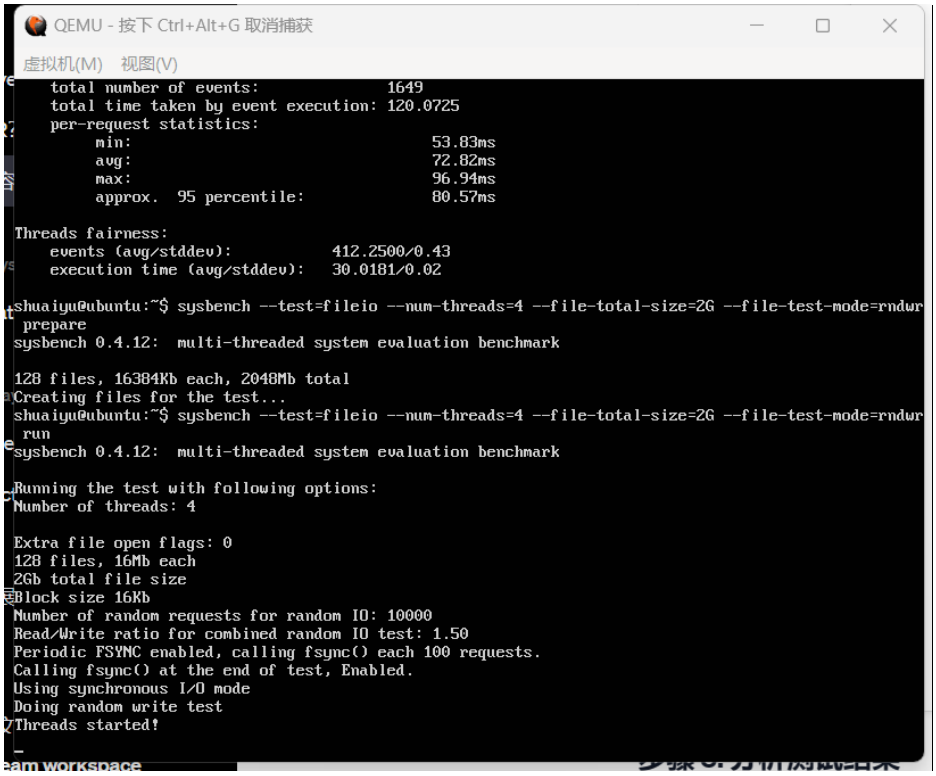
2. FILE I/O test:

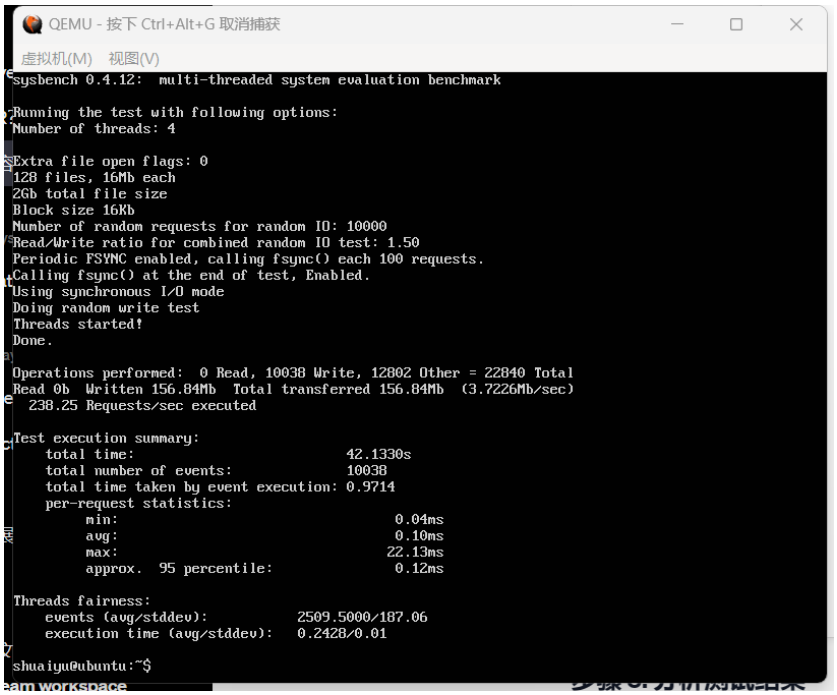
a. Test 3:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



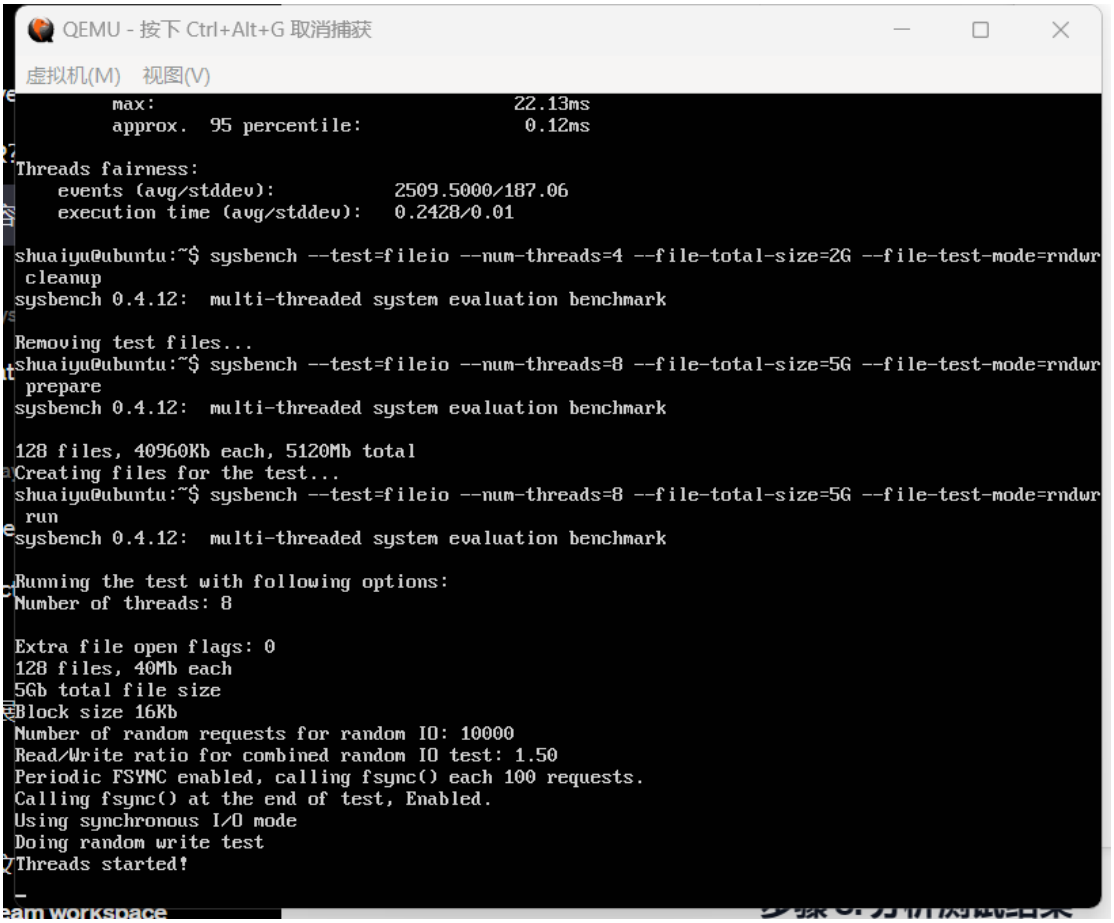


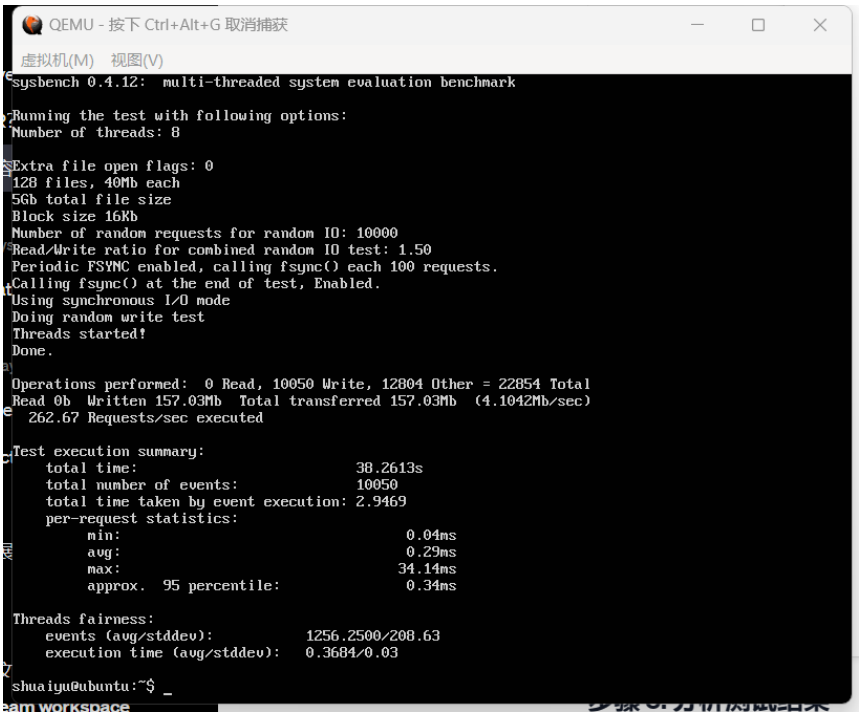
b. Test 4:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

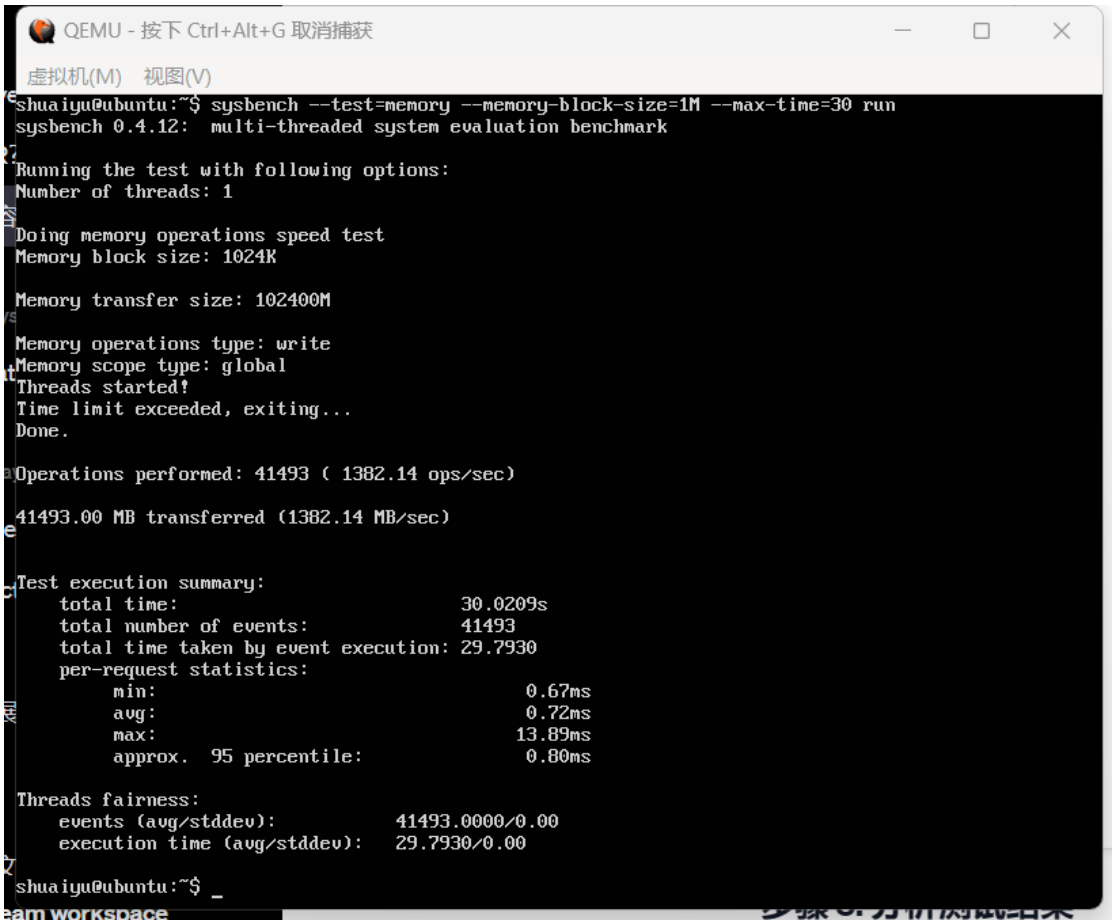
sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup





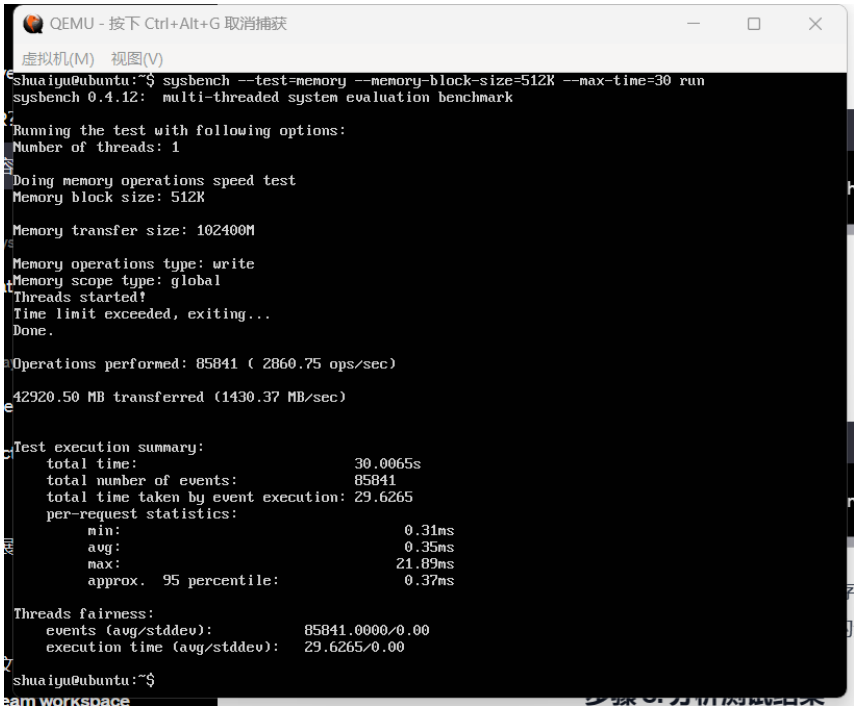
3. Memory test:
    a. Test 3:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 4:

sysbench memory --memory-block-size=512K --time=30 run

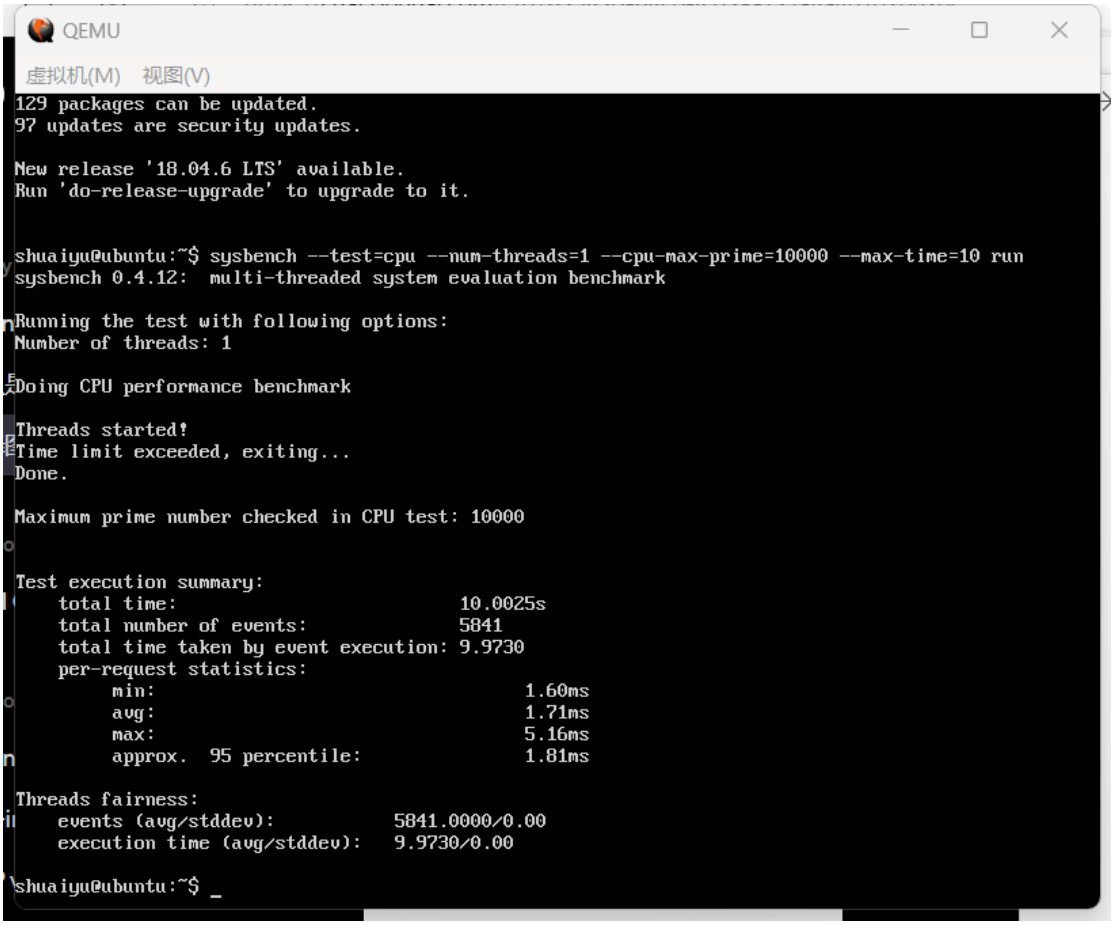


Environment 3:

1. CPU test:

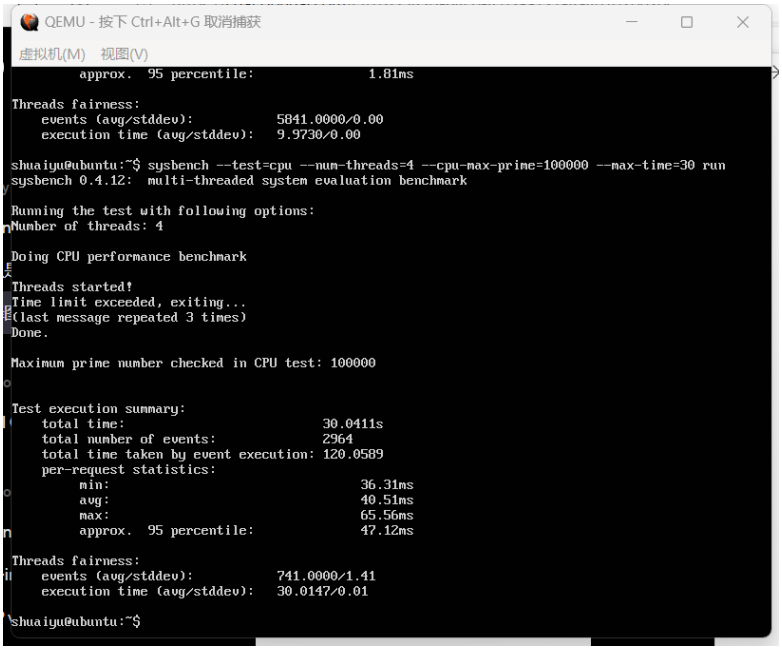
   a. Test 5:

   sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



   b. Test 6:

   sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



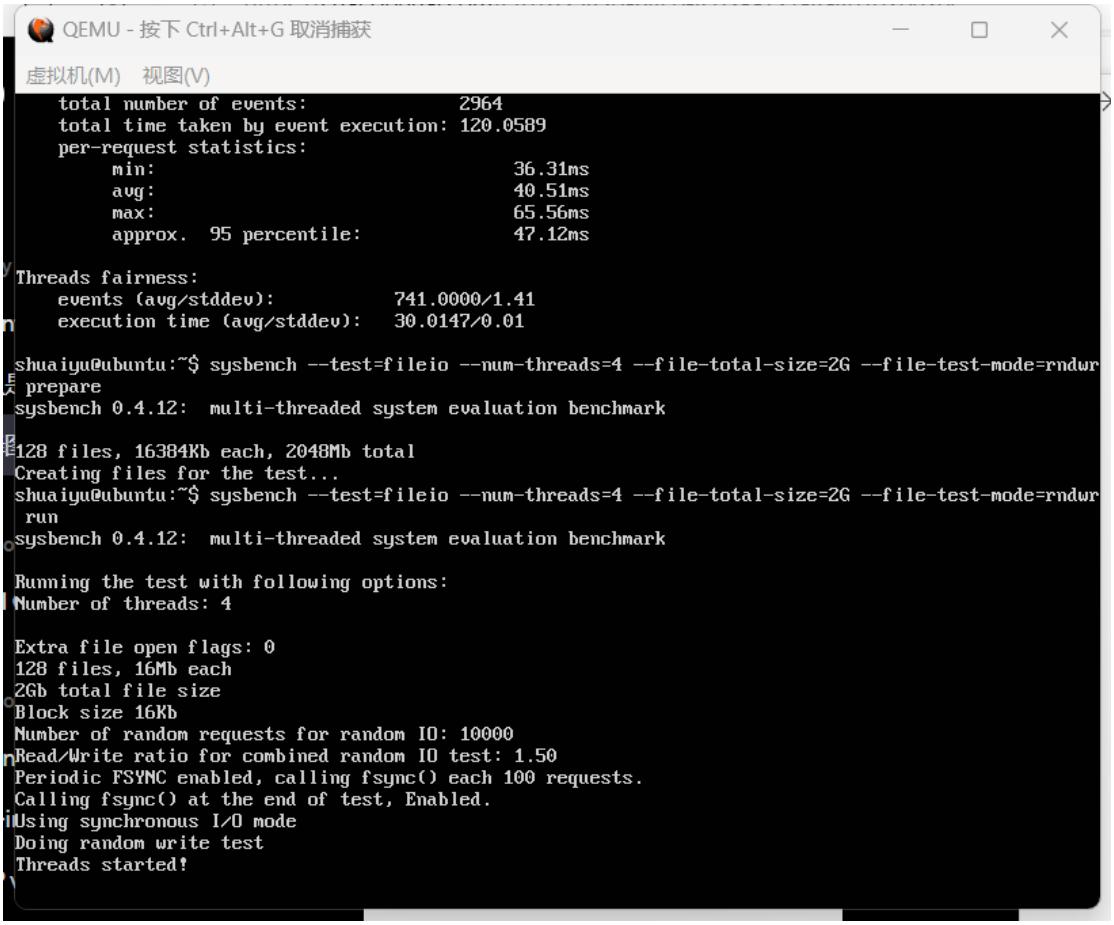
2. FILE I/O test:

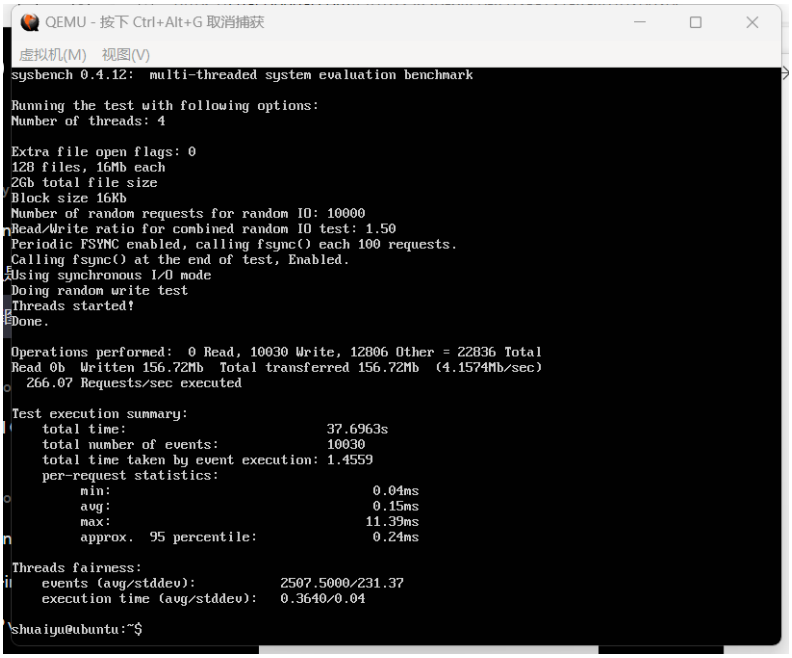
a. Test 5:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



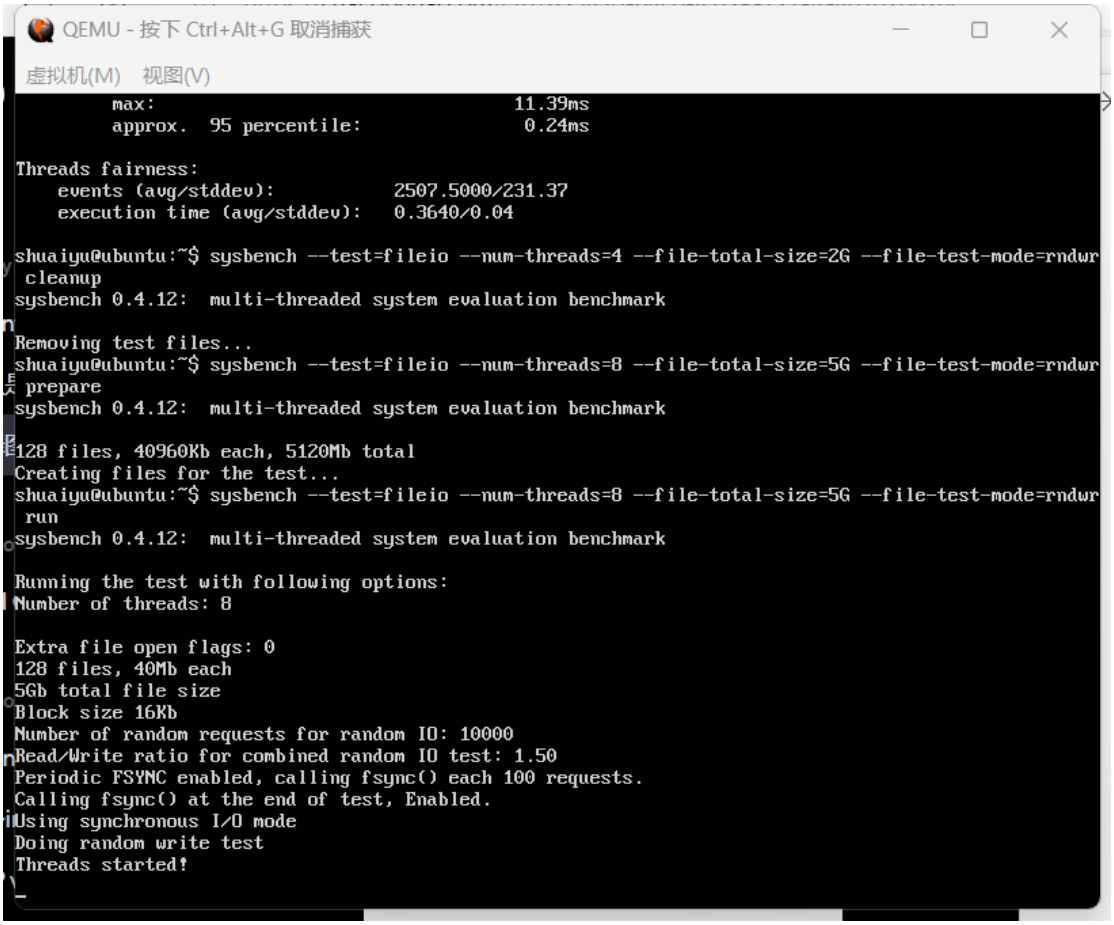


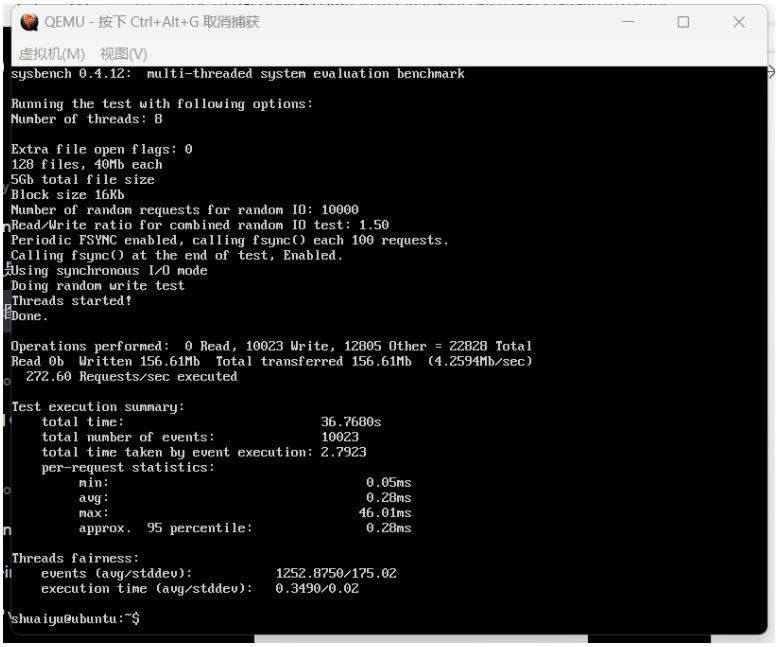
b. Test 6:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup

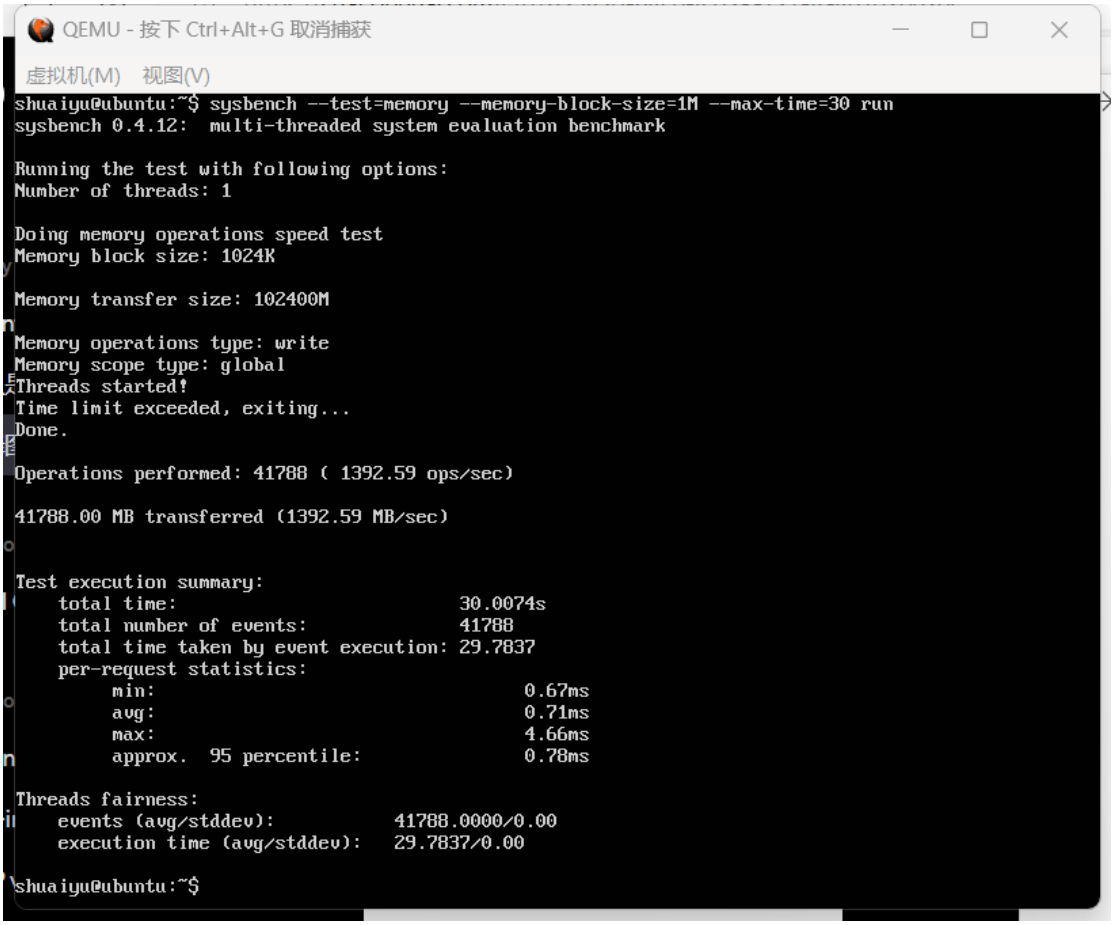




3. Memory test:

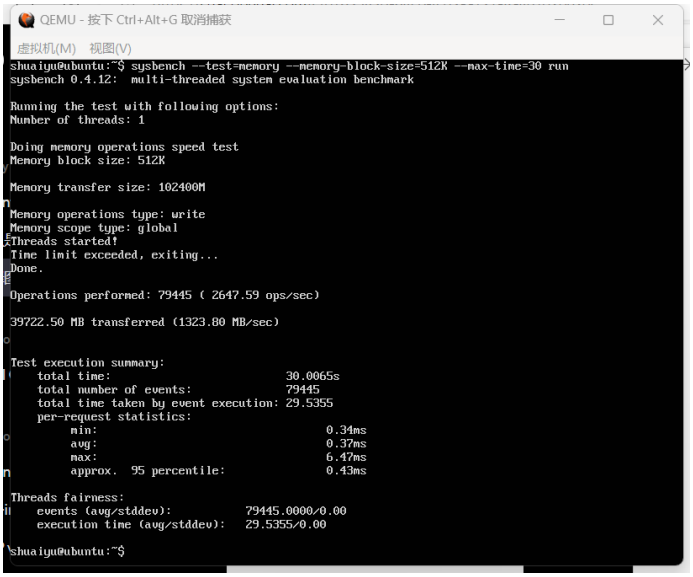
a. Test 5:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 6:

sysbench memory --memory-block-size=512K --time=30 run

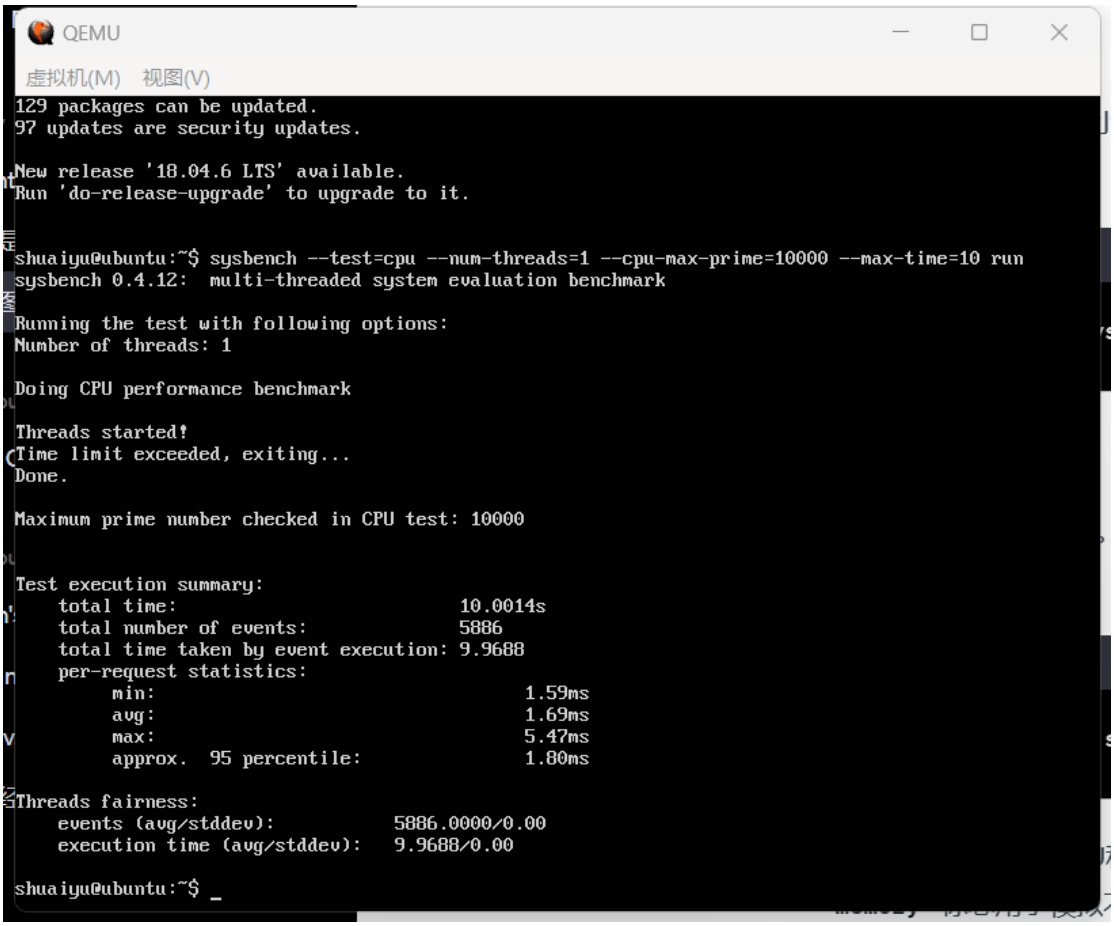


Environment 4:

1. CPU test:

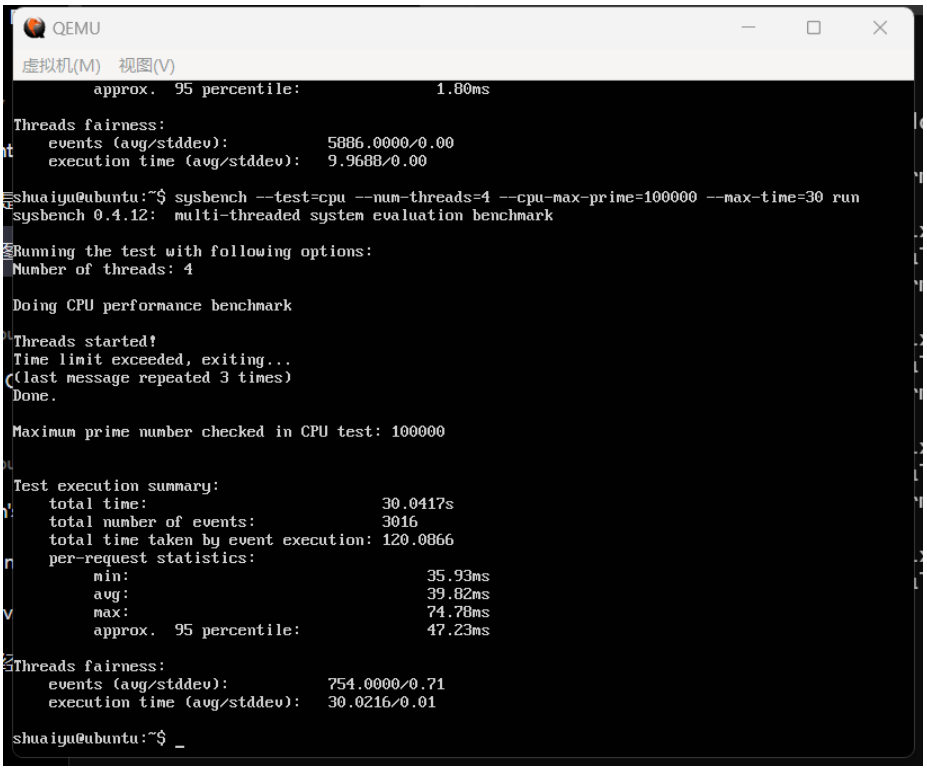
a. Test 7:

sysbench --test=cpu –num-threads=1 --cpu-max-prime=10000 --max-time=10 run



b. Test 8:

sysbench --test=cpu –num-threads=4 --cpu-max-prime=100000 --max-time=30 run



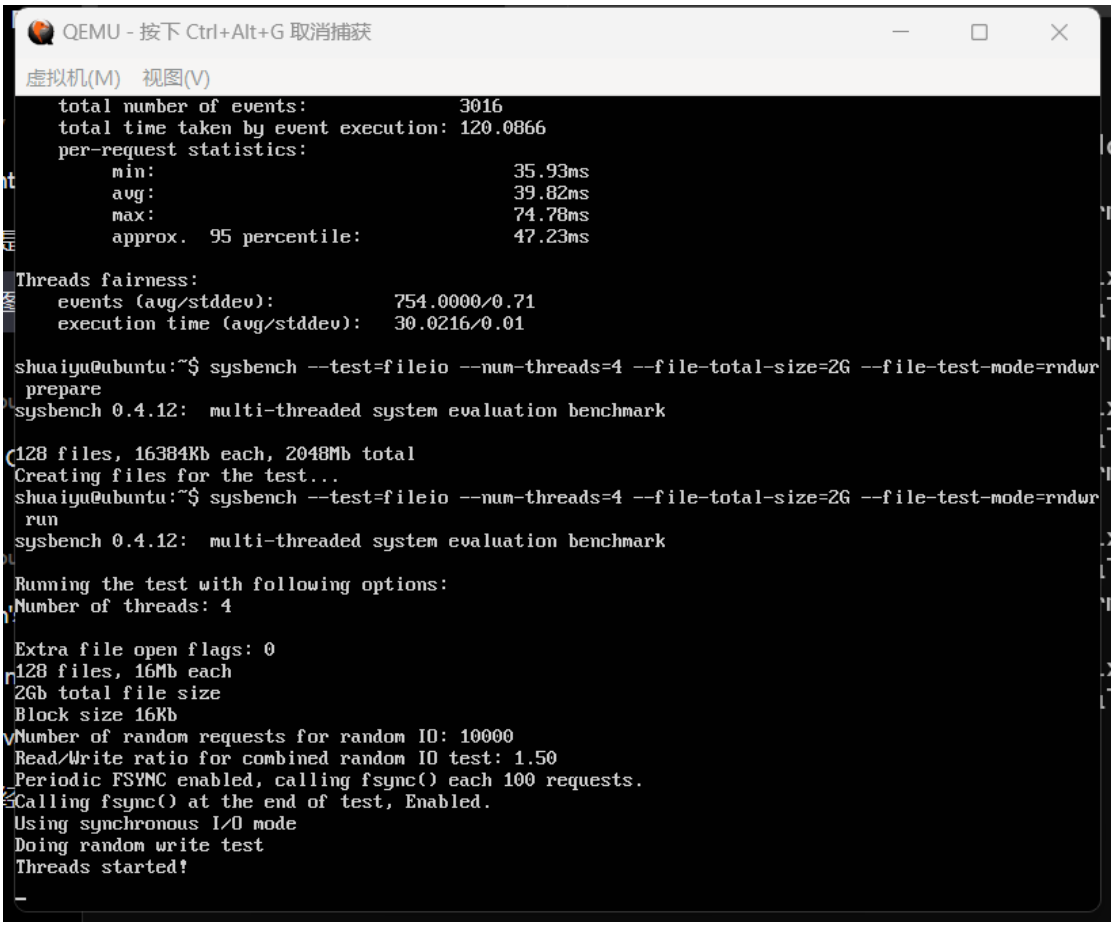
2. FILE I/O test:

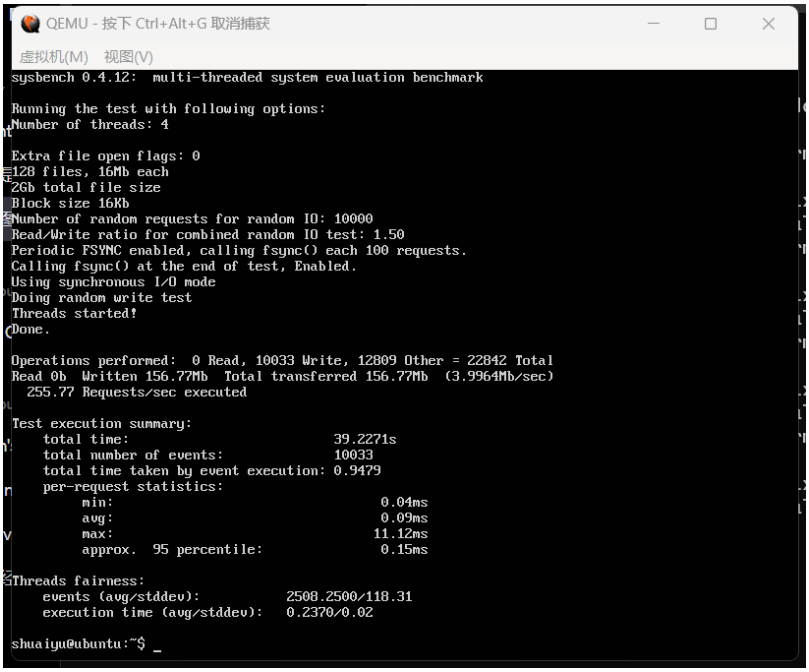
a. Test 7:

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

sysbench --num-threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup



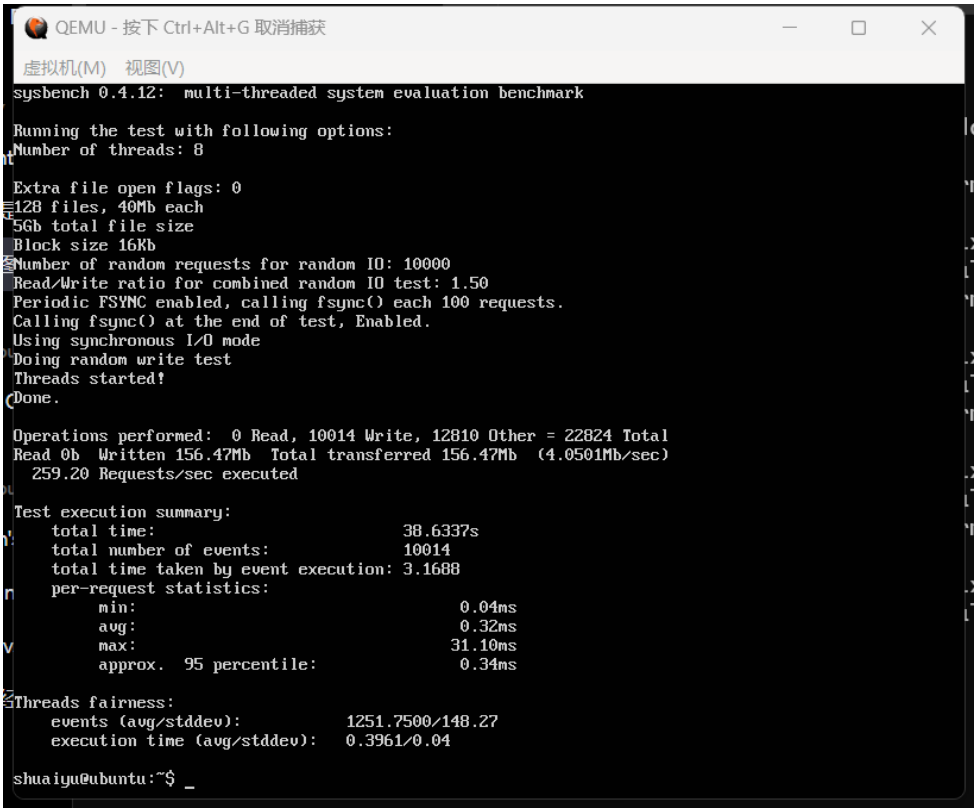


b. Test 8:

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr prepare

sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr run

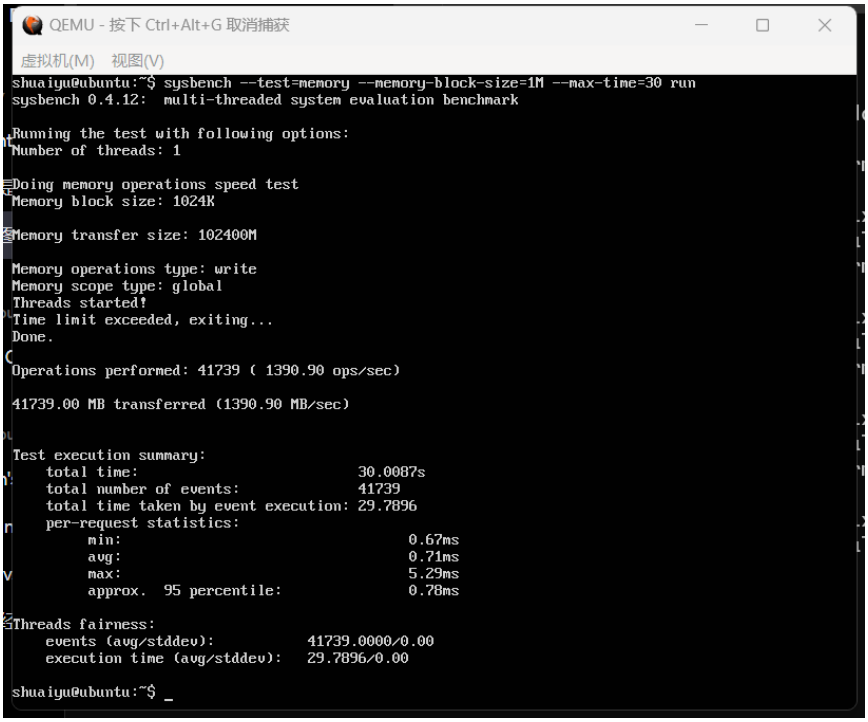
sysbench --num-threads=8 fileio --file-total-size=5G --file-test-mode=rndwr cleanup



4. Memory test:

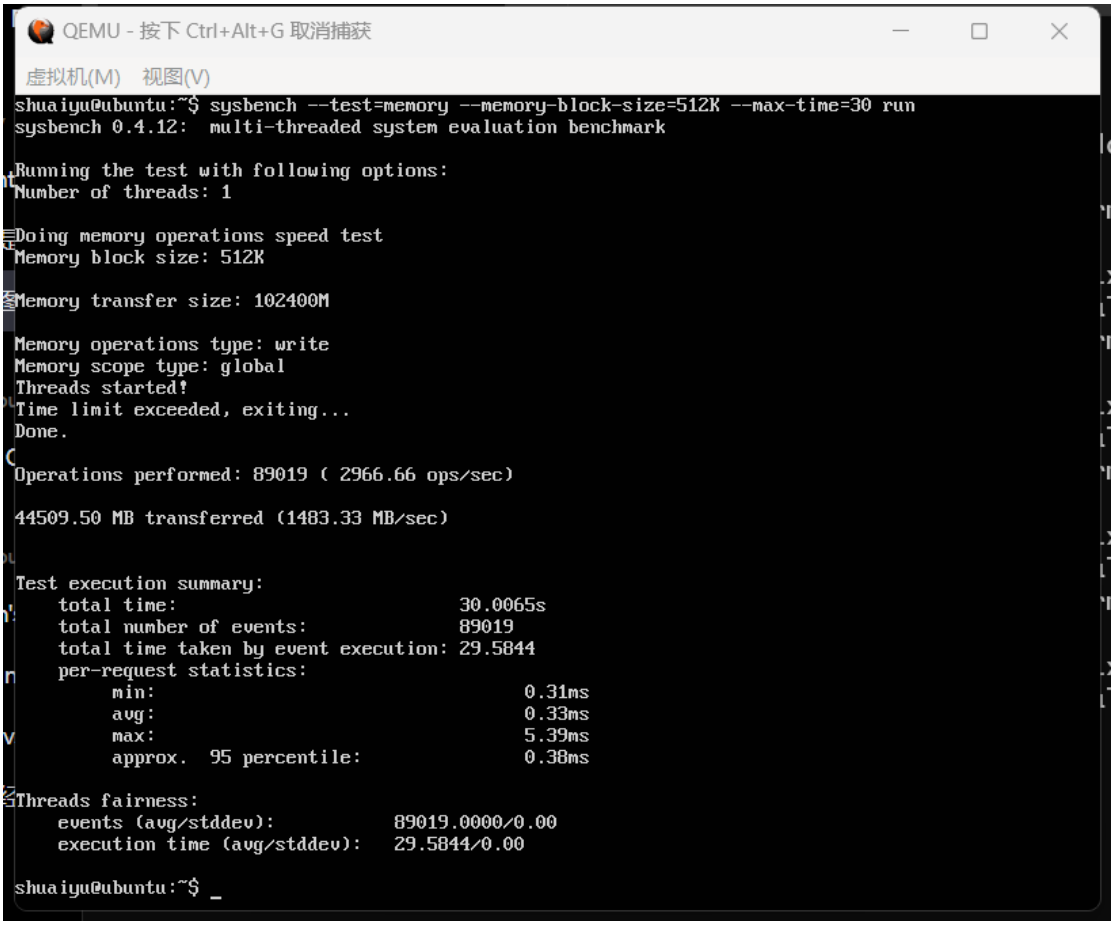
a. Test 7:

sysbench memory --memory-block-size=1M --time=30 run



b. Test 8:

sysbench memory --memory-block-size=512K --time=30 run



Analysis of Results

CPU Test:

Following parameters are considered while performing the cpu test.

a. --cpu-max-prime: This parameter sets the maximum prime number to check for in the CPU tests.

b. --time: This parameter sets the total time for the test in seconds.

c. --threads: This parameter sets the number of parallel threads to use in the test.

The experiment is done by altering the values of the above parameters and running each test case 5 times to obtain the average, min, max and std. values of the results.

Test 1:

sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run

Platform	Image	Environment	Min	Avg	Max	Std

Docker	ubuntu	1	0.69	0.72	1.59	0.67
Docker	ubuntu	2	0.69	1.42	51.15	23.37
Docker	ubuntu	3	0.69	1.43	51.27	61.70
Docker	ubuntu	4	0.69	0.72	1.58	0.40
QEMU	ubuntu.jmp	1	1.58	1.70	12.74	5.78
QEMU	ubuntu.jmp	2	1.60	1.72	5.23	1.66
QEMU	ubuntu.jmp	3	1.59	1.71	11.86	5.12
QEMU	ubuntu.jmp	4	1.59	1.70	5.06	1.73
QEMU	disk1.jmp	1	1.59	1.70	5.42	1.92
QEMU	disk1.jmp	2	1.59	1.69	5.04	1.73
QEMU	disk1.jmp	3	1.60	1.71	5.16	1.80
QEMU	disk1.jmp	4	1.59	1.69	5.47	1.93

Test 2:
sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run

Platform	Image	Environment	Min	Avg	Max	Std
Docker	ubuntu	1	16.75	69.72	181.44	78.11
Docker	ubuntu	2	16.71	136.71	199.78	87.25
Docker	ubuntu	3	16.75	137.17	199.96	87.58
Docker	ubuntu	4	16.78	70.20	303.76	133.56
QEMU	ubuntu.jmp	1	47.01	72.50	92.12	21.19
QEMU	ubuntu.jmp	2	45.18	72.84	101.27	27.32
QEMU	ubuntu.jmp	3	35.94	40.42	137.21	46.99
QEMU	ubuntu.jmp	4	35.37	39.54	71.69	18.19
QEMU	disk1.jmp	1	47.25	73.55	100.15	27.34
QEMU	disk1.jmp	2	53.83	72.82	96.94	21.47
QEMU	disk1.jmp	3	36.31	40.51	65.56	14.96
QEMU	disk1.jmp	4	35.93	39.82	74.78	19.98

FileIO Test:
Following parameters are considered to perform FileIO Test
a. --file-total-size: This parameter sets the total size of the test file in megabytes. --file-test-mode: This parameter sets the type of file I/O test to run (e.g. "seqwr", "seqrewr").
b. --threads: This parameter sets the total number of threads used for the process.
The experiment is done by altering the values of the above parameters and running each test case 5 times to obtain the average, min, max and std. values of the results.

Test 1:
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run

Platform	Image	Environment	Min	Avg	Max	Std
Docker	ubuntu	1	0.00	0.35	76.83	39.45
Docker	ubuntu	2	0.00	0.34	26.71	13.43
Docker	ubuntu	3	0.00	0.34	24.47	12.80

Docker	ubuntu	4	0.00	0.30	94.34	49.60
QEMU	ubuntu.jmp	1	0.04	0.16	15.57	7.73
QEMU	ubuntu.jmp	2	0.04	0.10	22.44	10.66
QEMU	ubuntu.jmp	3	0.05	0.15	10.15	4.72
QEMU	ubuntu.jmp	4	0.04	0.09	9.26	4.09
QEMU	disk1.jmp	1	0.04	0.19	38.18	18.54
QEMU	disk1.jmp	2	0.04	0.10	22.13	10.34
QEMU	disk1.jmp	3	0.04	0.15	11.39	5.68
QEMU	disk1.jmp	4	0.04	0.09	11.12	5.55

Test 2:

sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run

Platform	Image	Environment	Min	Avg	Max	Std
Docker	ubuntu	1	0.01	0.70	58.79	27.55
Docker	ubuntu	2	0.01	0.82	223.85	105.33
Docker	ubuntu	3	0.01	0.89	99.64	46.73
Docker	ubuntu	4	0.01	0.66	54.70	25.63
QEMU	ubuntu.jmp	1	0.05	0.42	53.88	25.29
QEMU	ubuntu.jmp	2	0.04	0.37	54.08	25.40
QEMU	ubuntu.jmp	3	0.05	0.29	45.46	21.35
QEMU	ubuntu.jmp	4	0.04	0.34	50.07	23.51
QEMU	disk1.jmp	1	0.05	0.40	48.73	22.87
QEMU	disk1.jmp	2	0.04	0.29	34.14	16.02
QEMU	disk1.jmp	3	0.05	0.28	46.01	21.61
QEMU	disk1.jmp	4	0.04	0.32	31.10	14.58

Memory Test:

Following parameters are considered to perform Memory Test

a. --memory-block-size: This parameter sets the block size of the test memory in megabytes.

The experiment is done by altering the values of the above parameters and running each test case 5 times to obtain the average, min, max and std. values of the results.

Test 1:

sysbench memory --memory-block-size=1M --time=30 run

Platform	Image	Environment	Min	Avg	Max	Std
Docker	ubuntu	1	0.04	0.04	0.35	0.15
Docker	ubuntu	2	0.04	0.09	50.37	25.56
Docker	ubuntu	3	0.04	0.09	50.35	25.54
Docker	ubuntu	4	0.04	0.05	0.35	0.15
QEMU	ubuntu.jmp	1	0.61	0.65	11.69	1.90
QEMU	ubuntu.jmp	2	0.66	0.73	13.32	1.52
QEMU	ubuntu.jmp	3	0.61	0.65	15.38	2.84

QEMU	ubuntu.jmp	4	0.67	0.72	75.47	84.53
QEMU	disk1.jmp	1	0.61	0.68	232.48	81.95
QEMU	disk1.jmp	2	0.67	0.72	13.89	3.00
QEMU	disk1.jmp	3	0.67	0.71	4.66	0.27
QEMU	disk1.jmp	4	0.67	0.71	5.29	0.46

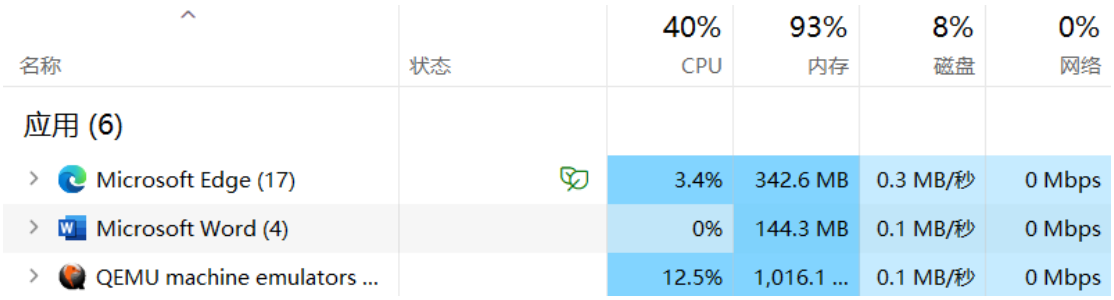
Test 2:
sysbench memory --memory-block-size=512K --time=30 run

Platform	Image	Environment	Min	Avg	Max	Std
Docker	ubuntu	1	0.02	0.02	0.29	0.09
Docker	ubuntu	2	0.02	0.04	50.33	25.56
Docker	ubuntu	3	0.02	0.04	50.39	25.57
Docker	ubuntu	4	0.02	0.02	0.34	0.09
QEMU	ubuntu.jmp	1	0.31	0.32	4.24	1.33
QEMU	ubuntu.jmp	2	0.31	0.34	8.68	2.18
QEMU	ubuntu.jmp	3	0.30	0.35	22.83	8.93
QEMU	ubuntu.jmp	4	0.31	0.33	13.62	3.39
QEMU	disk1.jmp	1	0.31	0.34	7.02	3.19
QEMU	disk1.jmp	2	0.31	0.35	21.89	8.77
QEMU	disk1.jmp	3	0.34	0.37	6.47	3.59
QEMU	disk1.jmp	4	0.31	0.33	5.39	2.38

Performance:
In a Windows 11 system, we can check the performance by typing CTRL+SHIFT+ESC to open the Windows Task Manager.
Performance data of QEMU:

CPU Memory FileIO

Image 1:
(1) During CPU test:



(2) During File IO test:

名称	状态	23% CPU	94% 内存	48% 磁盘	0% 网络
应用 (6)					
Microsoft Edge (16)		0%	236.1 MB	0 MB/秒	0 Mbps
Microsoft Word (4)		0%	111.2 MB	0 MB/秒	0 Mbps
QEMU machine emulators ...		15.5%	1,919.9 ...	8.1 MB/秒	5.3 Mbps

(3) During Memory test:

名称	状态	37% CPU	95% 内存	5% 磁盘	0% 网络
应用 (6)					
Microsoft Edge (15)		0%	153.4 MB	0.1 MB/秒	0 Mbps
Microsoft Word (4)		0%	146.1 MB	0.1 MB/秒	0 Mbps
QEMU machine emulators ...		16.0%	2,069.2 ...	0.3 MB/秒	0 Mbps

Image 2:

(1) During CPU test:

名称	状态	33% CPU	89% 内存	1% 磁盘	0% 网络
应用 (6)					
Microsoft Edge (16)		0%	211.6 MB	0 MB/秒	0 Mbps
Microsoft Word (4)		0%	159.4 MB	0 MB/秒	0 Mbps
QEMU machine emulators ...		11.5%	1,505.4 ...	0 MB/秒	0 Mbps

(2) During File IO test:

名称	状态	31% CPU	90% 内存	61% 磁盘	0% 网络
应用 (6)					
Microsoft Edge (15)		0%	103.1 MB	0 MB/秒	0 Mbps
Microsoft Word (4)		0.3%	76.2 MB	0.1 MB/秒	0 Mbps
QEMU machine emulators ...		27.9%	2,310.7 ...	4.2 MB/秒	0 Mbps

(3) During memory test:

名称	状态	43% CPU	93% 内存	17% 磁盘	0% 网络
应用 (6)					
Microsoft Edge (15)		0.9%	165.7 MB	0.1 MB/秒	0 Mbps
Microsoft Word (4)		0.9%	141.8 MB	0.1 MB/秒	0 Mbps
QEMU machine emulators ...		26.1%	2,224.4 ...	0.2 MB/秒	0 Mbps

Performance data of Docker:

(1) During CPU test:

名称	状态	22% CPU	87% 内存	2% 磁盘	0% 网络
应用 (6)					
Docker Desktop (4)		0.1%	99.5 MB	0 MB/秒	0 Mbps

(2) During File IO test:

名称	状态	36% CPU	93% 内存	71% 磁盘	0% 网络
应用 (6)					
Docker Desktop (4)		0%	66.5 MB	0.1 MB/秒	0 Mbps

(3) During Memory test:

名称	状态	17% CPU	92% 内存	3% 磁盘	0% 网络
应用 (6)					
Docker Desktop (4)		0%	79.7 MB	0 MB/秒	0 Mbps

For CPU Test:

CPU utilization in case of QEMU image 1 was around 12.5%, image 2 was around 11.5% whereas in case of Docker it was 0.1%.

For File I/O Test:

The FileIO utilization of QEMU image 1 was around 48%, image 2 was around 61% whereas in case of Docker it was 71%.

For Memory Test:

The memory utilization of QEMU image 1 was around 95%, image 2 was around 93%, whereas in case of Docker it was 92%.

Conclusion:

From the above CPU, file I/O and memory performance data, we can easily see that the performance of Docker is much higher than that of QEMU. Taking performance into consideration only, we can conclude that containers are a better choice for virtualization than virtual machines.

Git Repository Information:
https://github.com/StormYuchen/CSEN241-HMs.git

References

[1] QEMU Documentation: https://www.qemu.org/docs/master/
[2] Homebrew: https://brew.sh/
[3] How do I run QEMU on Windows: https://linuxhint.com/qemu-windows/
[4] Sysbench manual: https://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf
[5] Docker CLI Reference: https://docs.docker.com/engine/reference/commandline/cli/
[6] Docker exec: https://docs.docker.com/engine/reference/commandline/exec/
[7] Docker Tutorials: https://docker-curriculum.com/