Assignment 2: POSIX Programming & Benchmarking

1 Host Environment

1

2

3

5

6

10

11

For my host environment, I chose to run Ubuntu Server 24.04.2 LTS using a VirtualBox hypervisor. I chose this operating system as I have sufficient Linux experience to feel confident using an operating system with no graphical interface (as opposed to Ubuntu Desktop), and the absence of a GUI means a smaller ISO file, memory footprint, & CPU footprint. I chose Ubuntu specifically because it's a Linux system with which I have previous experience, and is well-document with plenty of packages available to install if needs be. Ubuntu also makes it easy to install the PREEMPT_RT patches, which transform the standard Linux kernel into a fully preemptible, real-time kernel, which I felt was more suitable for this assignment, as the standard Linux kernel is not suitable for a hard real-time system due to its lack of preemption.



Figure 1: Virtual machine hardware configuration

I set the virtual machine to have a single CPU and set the amount of RAM to 2048MB which is the recommended minimum for Ubuntu Server¹. I left the hard disk size at the default of 25GB as I saw no reason to change it. The real-time kernel with the PREEMPT_RT patches installed is available with Ubuntu Pro, which is free for personal use. After setting up an Ubuntu Pro account, I enabled the real-time kernel using the pro command.



Figure 2: Enabling the real-time kernel with the pro command

Finally, I transferred over the following C file (taken from the lecture slides) via scp to the virtual machine to get the clock resolution, which is 1 nanosecond:

```
#include<unistd.h>
#include<time.h>
#include <stdio.h>

int main(){
    struct timespec clock_res;
    int stat;
    stat=clock_getres(CLOCK_REALTIME, &clock_res);
    printf("Clock resolution is %d seconds, %ld nanoseconds\n",clock_res.tv_sec,clock_res.tv_nsec);
    return 0;
}
```



Figure 3: Getting the clock resolution of the virtual machine

2 CPU & Data-Intensive Applications

1

2

3

6

10

11

12

13

14 15

16

17

18 19

20 21

22

23

24 25

26 27

28

29 30

31 32

33

34

35

36

37 38

39

40

41

42

43

44

To develop my CPU & data-intensive programs, I chose to use Python for ease of development (and because any Python program will stress your CPU & memory no matter how simple \textcircled). I chose htop as my resource-monitoring tool as I have often used it in the past, it has easy to read & understand output, and shows you exactly what proportion of the CPU & memory is in use at that time. It also allows you to list processes by CPU consumption or memory consumption which is a useful option to have for this assignment.

```
import multiprocessing
import time
import argparse
import os
def stress_cpu(workload: float):
    .....
    Function to create CPU load. Uses a busy-wait method to simulate CPU usage.
    :param workload: The fraction of time (0.0 to 1.0) the CPU should be busy.
    .....
    cycle_time = 0.1 # Total cycle time (100ms per iteration)
    busy_time = cycle_time * workload # Time to stay busy
    idle_time = cycle_time - busy_time # Time to stay idle
    while True:
        start_time = time.time()
        while (time.time() - start time) < busy time:</pre>
            pass # Busy wait
        time.sleep(idle_time) # Sleep to control CPU usage
def start_stress_test(load: str):
    .....
    Starts CPU stress test based on load level.
    :param load: 'medium' (~50% load) or 'high' (~100% load)
    num_cores = os.cpu_count() or 4 # Use all available CPU cores
    workload = 0.5 if load == "medium" else 1.0 # Set workload percentage
    print(f"Starting {load.upper()} CPU stress test on {num_cores} cores...")
    processes = []
    for _ in range(num_cores):
        p = multiprocessing.Process(target=stress_cpu, args=(workload,))
        p.start()
        processes.append(p)
    try:
        for p in processes:
            p.join()
    except KeyboardInterrupt:
       print("Stopping stress test...")
        for p in processes:
```

```
p.terminate()
45
                 p.join()
46
47
    if __name__ == "__main__":
48
         parser = argparse.ArgumentParser(description="CPU Stress Test Script")
49
         parser.add_argument("--load", choices=["medium", "high"], required=True, help="Choose CPU load level
50
         \hookrightarrow (medium or high)")
         args = parser.parse_args()
51
52
         start_stress_test(args.load)
53
```

Listing 1: stress_cpu.py

CPU[52.3%] Tasks: 27, 36 thr, 86 kthr; 1 running 183M/1.92G] Load average: 1.37 1.00 0.52 0K/0K] Uptime: 00:07:35
Main I/O									
PID USER	PRI	NI	VIRT	RES	SHR S	CPU%	MEM%	TIME+	Command
1 root	20	- 0	22044	13200	9488 S	0.0	0.7	0:01.55	/sbin/init splash noprompt noshell automatic-ubiquity
307 root	19		66756	17552	16400 S		0.9	0:00.32	├ /usr/lib/systemd/systemd-journald
356 root	RT		282M	<mark>27</mark> 264	8704 S		1.4	0:00.07	├ /sbin/multipathd -d -s
369 root	20			<mark>27</mark> 264	8704 S		1.4	0:00.00	/sbin/multipathd -d -s
370 root	RT			<mark>27</mark> 264	8704 S	0.0	1.4	0:00.00	/sbin/multipathd -d -s
371 root	RT		282M	<mark>27</mark> 264	8704 S	0.0	1.4	0:00.00	/ /sbin/multipathd -d -s
372 root	RT		282M	27264	8704 S	0.0	1.4	0:00.00	/ /sbin/multipathd -d -s
373 root	RT		282M	27264	8704 S	0.0	1.4	0:00.02	/sbin/multipathd -d -s
374 root	RT			27264	8704 S	0.0	1.4	0:00.00	└──/sbin/multipathd -d -s
380 root	20		29144	7808	4864 S	0.7	0.4	0:00.43	
1190 root	20		29148	5124	2176 S	0.0	0.3	0:00.00	(udev-worker)
1191 root	20		29148	5124	2176 S	0.0	0.3	0:00.00	(udev-worker)
434 systemd-ne	20		18992	9600	8448 S	0.0	0.5	0:00.10	/usr/lib/systemd/systemd-networkd
469 systemd-re	20		21584	12800	10624 S	0.0	0.6	0:00.15	/usr/lib/systemd/systemd-resolved
4/1 systemd-ti	20		91020	7680	6784 S	U.U	0.4	0:00.09	/usr/11D/systemd/systemd-timesyncd
570 systemd-ti	20		91020	7680	6784 S	0.0	0.4	0:00.00	/usr/lib/systemd/systemd-timesyncd
598 messagebus	20		9808	5632	4736 S	U.U	0.3	0:00.09	dous-daemonsystemaddress=systemd:notorknopidtilesystemd-activationsyste
604 polkita	20		374M	9752	7424 S	0.0	0.5	0:00.09	/usr/lib/polkit-l/polkitdno-debug
704 polkita	20		374M	9752	7424 5	0.0	0.5	0:00.00	/usrviiD/polkit-i/polkitono-debug
705 pulkitu	20		374M	9752	7424 5	0.0	0.5	0:00.00	VISTV110/DOIRT-1/DOIRTONO-DEDUg
706 polkita	20		374M	9752	7424 5	0.0	0.5	0:00.00	/usp/iid/poikit-1/poikitano-debug
508 PUUT	20		1205M	10084	6784 5	0.0	0.5	0:00.31	/ Snap/canonical-il/vepatch/286/canonical-il/vepatch/
753 ruut	20		1205M	10004	6704 3	0.0	0.5	0:00.00	/shap/canonical-livepatch/266/canonical-livepatchd
756 MUUT	20		1205M	10084	6784 5	0.0	0.5	0:00.00	/snap/canonical-livepatch/286/canonical-livepatchu
797 PUUt	20		1205M	10084	6704 5	0.0	0.5	0:00.00	/snap/camprical-livepatch/266/camprical-livepatchu
730 FUUt	20		12051	10004	6704 0	0.0	0.5	0.00.00	/ Snap/camprical livepatch/200/camprical livepatchu
901 poot	20		12056	10004	6704 3	0.0	0.5	0.00.00	/snap/camprical_livepatch/200/camprical_livepatchu
902 poot	20		1200H	10004	6704 0	0.0	0.5	0.00.00	/ shap/canonical-livepatch/200/canonical-livepatchd
610 root	20		17290H	32468	21632 9	0.0	1.6	0.00.00	/isp/lib/canad/anad
626 root	20		1729M	32408	21632 9	0.0 0.0	1.0	0.00.01	List / Li
628 root	20		1729M	32408	21632 5	0.0 0 0	1.6	0.00.01	- /usr/lib/shand/shand
629 root	20		1729M	32408	21632 S	а. а.а	1.6	0:00.00	- /usr/lib/shand/shand
656 root	20		1729M	32408	21632 S	ă ă	1.6	0:00.03	
657 root	20		1729M	32408	21632 S	ñ.ñ	1.6	0:00.00	
667 root	20		1729M	32408	21632 S	0.0	1.6	0:00.19	- /usr/lib/snand/snand
905 root	20		1729M	32408	21632 S	0.0	1.6	0:00.03	/usr/lib/snand/snand
621 root	20		18120	8832	7808 S	0.0	0.4	0:00.11	/usr/lib/systemd/systemd-logind
624 root	20		457M	13696	11648 S	0.0	0.7	0:00.10	/usr/libexer/udisks2/udisksd
644 root	20		457M	13696	11648 S	0.0	0.7	0:00.00	/usr/libexec/udisks2/udisksd
645 root	20		457M	13696	11648 S	0.0	0.7	0:00.00	/usr/libexec/udisks2/udisksd
647 root	20		457M	13696	11648 S	0.0	0.7	0:00.00	/usr/libexec/udisks2/udisksd
	0	In the second	- / 1 /	1 1 - 1	Concert C	and the second second	-	and a second second	

Figure 4: htop output when running python3 stress_cpu.py --load medium

CPU[100.0%] Tasks: 29, 44 thr, 87 kthr; 1 running 191M/1.92G] Load average: 1.42 1.08 0.58 0K/0K] Uptime: 00:08:37
Main I/O	DDT	MIT	VIDT	DEC	CUD (2 001	9 MEM9	TIMEL	Compand
1 root	20	0	22044	13200	9488 9	з ого з а	0 0 7	0.01 58	Commana /chin/init_snlash_poppompt_poshell_automatic_ubiquitu
307 root	19	-1	66756	17680	16528 3	s 0.	0 0.9	0:00.33	/ softwarf spites most one of the international design of the spites
356 root	ŔŤ		282M	27264	8704 9	ς ο.	0 1.4	0:00.07	/sbin/multinathd -d -s
369 root	20		282M	27264	8704 3		0 1.4	0:00.00	⊢ /sbin/multipathd -d -s
370 root	RT		282M	27264	8704 3		0 1.4	0:00.00	─ /sbin/multipathd -d -s
371 root	RT			27264	8704 3		0 1.4	0:00.00	— /sbin/multipathd -d -s
372 root	RT			27264	8704 3		0 1.4	0:00.00	— /sbin/multipathd -d -s
373 root	RT			27264	8704 3		0 1.4	0:00.02	— /sbin/multipathd -d -s
374 root	RT			27264	8704 3		0 1.4	0:00.00	└─ /sbin/multipathd -d -s
380 root	20		29144	7808	4 864 3		0 0.4	0:00.45	— /usr/lib/systemd/systemd-udevd
1222 root	20		29148	5124	2176 3		0.3	0:00.00	— (udev-worker)
1223 root	20		29148	5124	2176 S		0 0.3	0:00.00	L (udev-worker)
434 systemd-n	e 20		18 992	9600	8448 3		0.5	0:00.10	— /usr/lib/systemd/systemd-networkd
469 systemd-r	e 20		<mark>21</mark> 584	12800	10624 3		0.6	0:00.15	— /usr/lib/systemd/systemd-resolved
471 systemd-t	i 20		91020	7680	6784 S		0 0.4	0:00.09	/usr/lib/systemd/systemd-timesyncd
570 systemd-t	i 20		91020	7680	6784 3		0 0.4	0:00.00	└─ /usr/lib/systemd/systemd-timesyncd
598 messagebu	s 20		<mark>9</mark> 808	5632	4736 3		0.3	0:00.11	@dbus-daemonsystemaddress=systemd:noforknopidfilesystemd-activationsyslo
604 polkitd	20		374M	9752	7424 3		0 0.5	0:00.09	─ /usr/lib/polkit-1/polkitdno-debug
704 polkitd	20		374M	9752	7424 3		0.5	0:00.00	─ /usr/lib/polkit-1/polkitdno-debug
705 polkitd	20		374M	9752	7424 3		0.5	0:00.00	/usr/lib/polkit-1/polkitdno-debug
706 polkitd	20		374M	9752	7424 3		0.5	0:00.00	└usr/lib/polkit-1/polkitdno-debug
608 root	20		1205M	10084	6784 3		0 0.5	0:00.31	 /snap/canonical-livepatch/286/canonical-livepatchd
753 root	20		1205M	10084	6784 3	s 0.	0.5	0:00.00	 /snap/canonical-livepatch/286/canonical-livepatchd
756 root	20		1205M	10084	6784 3	50.	0 0.5	0:00.00	 /snap/canonical-livepatch/286/canonical-livepatchd
/97 root	20		1205M	10084	6784	5 0.	0 0.5	0:00.00	 /snap/canonical-livepatch/286/canonical-livepatchd
798 root	20		1205M	10084	6784 3	5 0.	0.5	0:00.00	 /snap/canonical-livepatch/286/canonical-livepatchd
800 root	20		1205M	10084	6784 3		0.5	0:00.00	/snap/canonical-livepatch/286/canonical-livepatchd
001 ruut	20		1205M	10004	6704 0		0 0.5	0:00.00	/snap/canonical-livepatch/200/canonical-livepatchu
640 poot	20		12000	22409	91699 0		0 0.0	0.00.00	/snap/canonical-itvepatch/200/canonical-itvepatchu
626 poot	20		1720M	99460	21002 0		0 1.0 0 1.0	0.00.05	Lus / Lus / Lib / Shapur Shapu
620 root	20		1720H	22400	21002 0		0 1.0	0:00.01	/usr/lib/shapu/shapu
629 root	20		1729M	32400	21632		0 1.0	0.00.00	/dsi/lib/shapu/shapu
656 poot	20		1720H	22400	21002 0		0 1.0	0.00.00	/usr/lib/shapu/shapu
657 root	20		1729M	32408	21632		0 1.0 0 1.6	0.00.03	/ (sr/lib/and/snand
667 root	20		1729M	32408	21632 9		0 1 6	0:00.01	- /usr/lib/anad/saand
905 root	20		1729M	32408	21632 9		о <u>1.0</u> 0 1.6	0:00.13	
621 root	20		18120	8960	7936 9		0 0.4	0:00.13	/usr/lib/sustemd/sustemd-logind
624 root	20		457M	13696	11648	s ő.	0 0.7	0:00.10	- /usr/libexec/udisks2/udisksd
644 root	20		457M	13696	11648		0 0.7	0:00.00	/usr/libexec/udisks2/udisksd
645 root	20		457M	13696	11648 3		0 0.7	0:00.00	— /usr/libexec/udisks2/udisksd
647 root	20		457M	13696	11648 3		0 0.7	0:00.00	─ /usr/libexec/udisks2/udisksd
Holp EScature	econo.	bE 4	Filton	Lint	E Cont	DUCT	ico E	Nico IE9	

Figure 5: htop output when running python3 stress_cpu.py --load high

1

2

3 4

5

6

7

10

11 12

13

14

15 16

17

18

19 20

21

22

23 24

25

26

27 28

29

30

31

32 33

```
import argparse
import time
import psutil
def stress_memory(target_usage: float):
    .....
   Stress the system memory to a given percentage.
   :param target usage: Target memory usage (0.0 to 1.0, where 1.0 is 100%)
   .....
   total_memory = psutil.virtual_memory().total # Get total RAM in bytes
   target_memory = int(total_memory * target_usage) # Calculate target memory size
   print(f"Total Memory: {total_memory / (1024**3):.2f} GB")
   print(f"Target Memory Usage: {target_memory / (1024**3):.2f} GB ({target_usage * 100:.0f}%)")
   try:
        memory_hog = [] # List to store allocated memory chunks
        chunk_size = 100 * 1024 * 1024 # Allocate in 100MB chunks
        while sum(len(chunk) for chunk in memory_hog) < target_memory:</pre>
            memory_hog.append(bytearray(chunk_size)) # Allocate memory
            time.sleep(0.1) # Small delay to allow system response
       print("Memory fully allocated. Holding...")
       while True: # Keep the memory occupied
            time.sleep(1)
   except MemoryError:
        print("Memory limit reached. Exiting...")
   except KeyboardInterrupt:
       print("Memory stress test stopped.")
```

```
34 if __name__ == "__main__":

35 parser = argparse.ArgumentParser(description="Memory Stress Test Script")

36 parser.add_argument("--usage", type=float, default=1.0, help="Target memory usage (default: 1.0 for

→ 100%)")

37 args = parser.parse_args()

38
```

stress_memory(args.usage)

39

Listing 2: stress_memory.py

CPU[Mem[3.3%] Tasks: 29, 47 thr, 87 kthr; 1 running .856/1.926] Load average: 1.25 1.24 0.87
Swp[0K/0K] Uptime: 00:15:57
Main	I/0								
PID+	USER	PRI	NI	VIRT	RES	SHR S	CPU%	MEM%	TIME+ Command
1 1	root	20	0	22044	7952	4240 S	0.0	0.4	0:01.65 /sbin/init splash noprompt noshell automatic-ubiquity
3071	root	19		66756	4880	3728 5	0.0	0.2	0:00.39 / Vush/IID/Systemd/Systemd-Journald
356	ruut	50		202M	27264	0704 5	0.0	1.4	0:00.08 /spin/huripathu -0 -s
307 1	rout	20		20211	27204	0704 0	0.0	1.4	a 20 chin/multipathd -u -s
370	ruut 200†	DT		2028	27204	0704 0	0.0	1.4	e e e
372	root	PT		2020	27204	9704 0	0.0	1.4	a a a a b chin/multipathd -d -s
372	root	PT		282M	27264	8704 S	0.0	1.4	0.00 S
374	root	RT		282M	27264	8704 9	0.0 0.0	1 4	a a a a chain multipathd -d -s
380	root	20		29144	6272	3328 5	а. а.а	й. Т	0:00.69 //usr/lib/sustemd/usremd-udevd
1792	root	20		29148	4996	2048 S	ă ă	й. 02	
1793	root	20		29148	5060	2048 S	<u>й.й</u>	й.З	
434		20		18992	4096	2944 S	0.0	0.2	0:00.10 //usr/lib/sustemd/sustemd-networkd
469		20		21584	5120	2944 S	0.0	0.3	0:00.16 //usr/lib/systemd/systemd-resolved
471		20		91020	4096	3200 S	0.0	0.2	0:00.09 //usr/lib/systemd/systemd-timesyncd
570		20		91020	4096	3200 S		0.2	0:00.00 /usr/lib/systemd/systemd-timesyncd
598		20		<mark>9</mark> 808	<mark>4</mark> 864	3968 S		0.2	0:00.14 🗕 @dbus-daemonsystemaddress=systemd:noforknopidfilesystemd-activationsysl
604	polkitd	20		374M	<mark>6</mark> 296	3968 S		0.3	0:00.09 🛏 /usr/lib/polkit-1/polkitdno-debug
704	polkitd	20		374M	<mark>6</mark> 296	3968 S		0.3	0:00.02 //////////////////////////////////
705		20		374M	6296	3968 S		0.3	0:00.00 // /usr/lib/polkit-1/polkitdno-debug
706		20			6296	3 968 S		0.3	0:00.00 // /usr/lib/polkit-1/polkitdno-debug
608		20		1205M	4836	1536 S		0.2	0:00.31 📂 /snap/canonical-livepatch/286/canonical-livepatchd
753 -		20		1205M	4836	1536 S		0.2	0:00.04 //////////////////////////////////
756		20		1205M	4836	1536 S		0.2	0:00.00 // /snap/canonical-livepatch/286/canonical-livepatchd
797 ו		20		1205M	4 836	1 536 S		0.2	0:00.00 // /snap/canonical-livepatch/286/canonical-livepatchd
798		20		1205M	4836	1536 S	0.0	0.2	0:01.04 /snap/canonical-livepatch/286/canonical-livepatchd
800 1	root	20		1205M	4836	1536 S	0.0	0.2	0:00.00 /snap/canonical-livepatch/286/canonical-livepatchd
801	root	20		1205M	4836	1536 S	0.0	0.2	0:00.00 /snap/canonical-livepatch/286/canonical-livepatchd
802 1	root	20		1205M	4836	1536 S	0.0	0.2	0:00.07 //snap/canonical-livepatch/286/canonical-livepatchd
610 1	root	20		1729M	14232	3456 S	0.0	0.7	0100.09 - Zust/2110/Snapd/Snapd
626 1	root	20		1729M	14232	3456 S	0.0	0.7	0100.02 /USP/TID/Shapu/Shapu
628	POUT	20		1729M	14232	3456 S	0.0	0.1	0.00.00 //III/Shapu/Shapu
629 1	ruut	20		1729M	14232	3456 8	0.0	0.7	
650 1	ruut	20		17200	14232	3430 3 9456 P	0.0	0.7	0:00.05 / /US//ID/Shapb/Shapb
667	root	20		1729M	14232	9456 C	0.0	0.7	4.00 Jan Jun Jun Jun Jun Jun Jun Jun Jun Jun Ju
905	root	20		1729M	14232	3456 9	0.0	0.7	4.00 C
621	root	20		18120	4480	3456 9	0.0	0.7	4:00 Ja / //is/ten/lig/sustemd-logind
624	root	20		457M	6144	4096 S	0.0	0.2 0.3	
644	root	20		457M	6144	4096 S	0.0	0.3	9:00.01 E /usr/libexer/udisks2/udisksd
645	root	20		457M	6144	4096 S	0.0	0.3	0:00.00 / /usr/libexec/udisks2/udisksd
647 1	root	20		457M	6144	4096 S	0.0	0.3	0:00.00 - /usr/libexec/udisks2/udisksd
F1Help	F2 <mark>Setup F3</mark> S	Searc	hF4	Filter	5List	F6Sort	By <mark>F7</mark> Nic	e - <mark>F</mark>	BNice +FSKill F10Quit

Figure 6: htop output when running python3 stress_memory.py --usage 0.85

References

[1] Canonical Group Ltd. *Basic Ubuntu Server Installation*. Accessed: 2025-03-18. 2025. URL: https://documentation.ubuntu.com/server/tutorial/basic-installation/.