

Assignment 2: POSIX Programming & Benchmarking

1 Host Environment

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-documented with plenty of packages available to install if needed. 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.

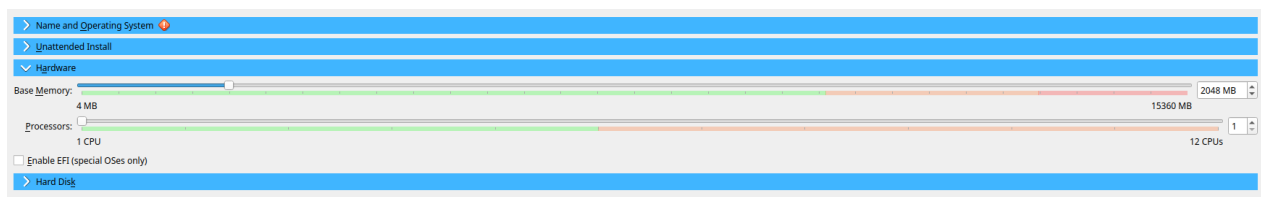


Figure 1: Virtual machine hardware configuration

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.

```
andrew-hayes@UbuntuRTS:~$ sudo pro enable realtime-kernel
One moment, checking your subscription first
Real-time kernel cannot be enabled with Livepatch.
Disable Livepatch and proceed to enable Real-time kernel? (y/N) y
No variant specified. To specify a variant, use the variant option.
Auto-selecting generic variant. Proceed? (y/N) y
Disabling incompatible service: Livepatch
Executing '/snap/bin/canonical-livepatch disable'
The Real-time kernel is an Ubuntu kernel with PREEMPT_RT patches integrated.

This will change your kernel. To revert to your original kernel, you will need
to make the change manually.

Do you want to continue? [ default = Yes ]: (Y/n) y
Configuring APT access to Real-time kernel
Updating Real-time kernel package lists
Updating standard Ubuntu package lists
Installing Real-time kernel packages
Real-time kernel enabled
A reboot is required to complete install.
andrew-hayes@UbuntuRTS:~$
```

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:

```
1 #include<unistd.h>
2 #include<time.h>
3 #include <stdio.h>
4
5 int main(){
6     struct timespec clock_res;
7     int stat;
8     stat=clock_getres(CLOCK_REALTIME, &clock_res);
9     printf("Clock resolution is %d seconds, %ld nanoseconds\n",clock_res.tv_sec,clock_res.tv_nsec);
10    return 0;
11 }
```

```

andrew-hayes@UbuntuRTS:~$ gcc -lrt res.c
res.c: In function 'main':
res.c:9:34: warning: format '%d' expects argument of type 'int', but argument 2 has type '__time_t' {aka 'long int'} [-Wformat=]
   9 |     printf("Clock resolution is %d seconds, %ld nanoseconds\n", clock_res.tv_sec, clock_res.tv_nsec);
     |                                ^~
     |                                |
     |                                int      __time_t {aka long int}
     |                                %ld
andrew-hayes@UbuntuRTS:~$ ./a.out
Clock resolution is 0 seconds, 1 nanoseconds
andrew-hayes@UbuntuRTS:~$

```

Figure 3: Getting the clock resolution of the virtual machine

References

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