Top Secret Inc: Technical Evaluation of an Operating System

Ivan Novasak

Southern New Hampshire University

IT 600: Operating Systems

Michael Gallaher

2024 April 28

# Top Secret Inc: Technical Evaluation of an Operating System

This document is to present an alternative general purpose operating system (OS) for Top Secret, Inc. that will replace their existing OS, TSI OS, which is suitable for embedding computing but not suitable for office computer work. The author presents aspects of general purpose operating systems that don’t exist in TSI OS and why a switch to a general purpose OS is ideal for the office, with an emphasis on the Linux OS.

# Organizational Needs and Requirements

The first section of this document is to cover the business requirements Top Secret, Inc. (TSI) has for their back-office, how different computing and OS concepts relate to those requirements, and why TSI OS is not suitable for office computing.

## Organizational Profile

The following general purpose operating system features will be explained in this section and showing why they are necessary in an office/business setting, including at Top Secret, Inc. (TSI):

* Multiprogramming;
* Multiprocessing;
* Multithreading;
* Virtual Memory;
* System Call Interface;
* Security;
* Device Drivers;
* Fault Tolerance.

***Multiprogramming***

Multiprogramming is the ability for a computer to run multiple programs at once via context switching, which is where the state of the active program is preserved when the user switches to another program (Sheldon, 2022). Multiprogramming allows the system to store multiple programs in memory, and the CPU executes portions of each very quickly, allowing for users to do multiple tasks at the same time.

Multiprogramming will allow a company to run multiple programs on the same computer. The computer will still need to switch from 1 task to the other if it has multiprogramming but not multiprocessing, but at least multiple programs can run via incorporating multiprogramming with more RAM to temporarily store the instructions of a program that is not actively being used. With multiprogramming, the users can use the computer for a variety of office tasks like email, word processing, spreadsheets, slide shows, instant messaging, video conferencing, and more. Many of these programs can be kept in the background in memory when they're not actively being used.

With only 1 program able to run at once, the computer becomes more like an appliance when it is running a program in that it can not be processing data in the background for other tasks/computers. A business restricted to such a computer would need N computers for N simultaneous tasks, where N is an integer, resulting in extremely high costs and inefficient use of processing and energy resources as well as still having the disadvantages of having a hard time getting the computers to exchange data amongst each other. One example of something that would not be easy to do is opening a web page and sending the link to it as an instant message to someone else, because the instant messenger would not be able to be run while the web browser was open and vice-versa. Anything that is best suited to be a background program/task would not be efficient on a single-task computer, such as a program to remind people of upcoming activities or appointments. A single-task computer would have to have that program open 24/7 just to remind the user of an upcoming event only a few times per day or week. So, it is clear that any business, including offices, needs a computer that can run multiple programs simultaneously.

***Multiprocessing***

Multiprocessing is the presence of multiple processors (CPUs) inside a single computer, which allow more than one program or process to run simultaneously (Yasar, 2023). A computer system can have any combination of multiple CPU chips with one ‘core’ per chip, one chip with multiple cores, or multiple chips with each having multiple cores (Yasar, 2023). The advantages of multiprocessing allow businesses to have the computer carry out many different office tasks at once, such as downloading new emails and checking for upcoming appointments in the background while the user is typing up a document and listening to music on the same computer. The user could also, for instance, be printing one document while writing or reading another. It is always good practice to have a dictionary around when reading and writing, and so multiprocessing would allow for the dictionary document and the user’s own document to be open simultaneously in case the user discovers a word they don’t understand. Even if the user is doing only 1 task, that task can be done faster with multiple CPUs because there would likely still need to be multiple processes operating and those would then execute simultaneously, allowing for increased throughput (Yasar, 2023). The vast majority of desktop and laptop CPUs sold and made are multicore, which is a type of CPU chip that has multiple processors inside of it.

Some computer applications consist of more than 1 program, and so multiprocessing is beneficial for such programs because it allows them to run. Otherwise, alternative applications would need to be used that each have only 1 program, perhaps causing a less efficient workflow. With multiprocessing, a person could, for instance, (A) have a web browser with Google Maps open showing directions to the office, (B) have an email program open to write those directions for a person, (C) be printing a form out for that person who will be arriving and, (D) logging the conversation in a notes profile for the client. The printing could be happening simultaneously thanks to multiprocessing, saving time at the office. Programs that do other features, such as phone calling and recording (where legal and allowed, with notice to all parties prior) could be technologically implemented simultaneously without worry when multiple processors are involved.

Without multiprocessing, the computer would have to keep switching its focus between the various processes in memory that the user has open, causing more waiting time as the computer needs to always ‘catch up’ with what the user was doing. Non-multiprocessing CPUs are also very uncommon these days and so it’s more likely that any on the market are older and without warranty support, which can be both a security and reliability risk for the business by not having current manufacturer support (Gallaher, 2024). Not having multiprocessing also will cause operations to take longer to perform, causing the business to have longer turnaround times for work requests. This will cost the business more both in wages and possible lost customers who will seek out businesses who do use multiprocessing computer systems.

***Multithreading***

Multithreading is the ability for a processor to carry out multiple instructions that are part of the same process, sharing that process’s address space (Kirvan, 2022). Multithreading will allow the CPUs to not need to open multiple instances of the same process, allowing for memory and CPU cycles to be preserved. One example where multithreading is in action is when a web browser is loading web pages with multiple pictures - each thread can handle the downloading and visual processing of the pictures. Multithreading is generally made for when multiple CPUs/cores need to work on the same process to accelerate the functionality of that process, as opposed to multiprocessing which is more ideal for multiple disparate processes/programs.

Multithreading allows the business to run a wider variety of applications, namely more complex applications like office suites and audio/video processing software, which either greatly benefit from both multiprocessing and multithreading, or require these processor features to run at all. Having multithreading will also allow tasks to finish faster because the processor can apply more clock cycles to the same task.

Multithreading will allow the business to use fewer overall applications instead of having a separate program or application for each individual task, possibly taking up less disk space and having an easier workflow with less bugs than the prior way would would involve separate programs for word processing, spreadsheet, slide shows, audio, video, printing, email, and web browsing. Some of these, namely word processing, spreadsheet, slide shows, could be part of the office suite application.

Not having multithreading will require all software used on the computer to be of the variety that never supported multiple threads. Most complex applications like office suites or video editors are designed with multiple threads in mind, and just won’t run at all on a single-threaded processor (Gallaher, 2024). The business impact is not having the latest supported software, which can lead to both security vulnerabilities and higher costs should something go wrong, as one is using software and/or equipment that are no longer supported.

***Virtual Memory***

Virtual memory is a reserved block of hard disk or solid state drive space that the computer uses when the operating system detects that physical memory (RAM) is running low (Gillis et al, 2022). Virtual memory is necessary, especially when running multiple large programs simultaneously that may take up the entire capacity of the RAM, as otherwise the computer will run out of RAM and therefore have no working memory for processes to execute, causing a crash (Gallaher, 2024). A computer that turns to virtual memory very often will run slower (known as ‘thrashing’) due to the hard disk or solid state drive being slower than RAM, and is a sign that more physical RAM is needed. Virtual memory will always be necessary no matter how much RAM one has, as one can not anticipate when a program or process will require more memory than the RAM that is installed in the system.

Virtual memory will allow programs that require more RAM than exists in the computer to continue running instead of crashing the OS when RAM runs out or low. This is the main benefit of virtual memory.

This will allow businesses to use computers for a bit longer before needing to add more RAM, however a notification feature should be implemented into the OS that tells the IT team if any computers are turning to virtual memory too much which is a sign more RAM is needed. Businesses even with enough RAM in computers can benefit from virtual memory because the virtual memory can be used to transfer data that isn’t being used much out of RAM and into the hard disk, allowing the faster, more expensive RAM to be used specifically for programs the user has in focus.

Not having virtual memory will cause the computer to crash (stop functioning) every time it runs out of physical RAM, which can mean hours of lost work if the work was not saved prior to the crash and lost income for any data that was lost due to the crash. Users would also need to monitor precisely how much RAM is available and reboot the computer frequently before the RAM is full, a process that can be several minutes for each reboot.

***System Call Interface***

The system call interface deals with how computer hardware interacts with user programs via process control, file management, device management, information maintenance, and communications between components (Williams, 2024). The system call interface takes commands the user enters and translates them into a form the kernel (core/main part of the operating system) can understand and communicate in Assembly or machine language to the hardware components. It is where processes are started from, ended, modified, and executed. This is also where the OS handles different types of hardware and configures the hardware for the OS’s support. This could be considered the heart of an operating system. According to Red Hat, the kernel in Linux is the primary component of the OS and it is responsible for memory management, process management, device drivers, system calls, and security (Red Hat, 2019). It is sandwiched between the physical hardware and the processes the user does on the computer (Red Hat, 2019).

A system call interface allows the users to have an easier-to-use computing experience, via the graphical user interface or even other features like voice command / digital assistants, as opposed to being forced to type commands all the time for basic computer / office tasks. This allows people who are not computer specialists to also use computers at TSI or any other company to get office and other tasks done without needing to worry about the underlying structure of how the computer works.

The type of system call interface chosen should be a form that is already standardised so as to not need to do what others have done. It is better to stand on the shoulder of giants, so something like a POSIX-compliant interface or even one of the Linux distributions (which are POSIX-compliant). MacOS, being based on BSD, could fit the bill. It is already something millions of people are familiar and comfortable with and has a wide support team in Apple Care.

Without a system call interface, there isn’t what might be considered an operating system, as then the computer would require hardware engineers to get the computer to do even the most basic functions like managing files, adding numbers, typing text, communicating with a network, or getting the computer to recognise a new hardware component. Every function or process would need to be developed from scratch and using it would require many different commands. This would be totally unacceptable in an office setting where people are expected to have computers run without the need for highly specialised hardware engineers around (a more general IT support employee though could help office employees with a computer with a full OS including system call interface). Such engineers could cost the company a lot of money for little practical benefit as the computer would still do minimal functions without a fully operational OS with today’s easy-to-use graphical features.

***Security***

Security in an operating system is essential. In this context, it is the use of techniques such as user authentication, 1-time passwords, firewalls, physical security, operating system policies, and maintenance/patches to ensure no unauthorised person can access or damage private data (JavaTpoint, n.d. a). Threats may include viruses, trojan horses, worms, program trap doors, network vulnerabilities, and social engineering. To protect against these threats a company’s IT team needs to keep abreast of new vulnerabilities, be in touch with the operating system and other software manufacturers in case patches to vulnerabilities are issued. Also, the IT team must talk to computer users about social engineering about proper computer usage, including changing passwords regularly, not giving passwords or other private data to unauthorised people.

Having a comprehensive security policy for applications ensures only specific authorised users can use them, and this security policy can be tied to data via encryption to ensure that even if some data fell into the wrong hands it can not be read.

Besides building security and physical security of the hardware, implementation of multi-factor authentication and encryption into the OS is vital. It may take a bit longer to access the data and applications’ features, but it is well worth it considering how much hacking, identity theft, and other malicious digital crime happens every day. The system could start by first asking for the user’s user name and password, then prompting for a code sent to their phone’s authentication app, then the user enters the value sent, and finally a fingerprint or facial scan is taken by the hardware and compared to the known stored captures at account setup time, finally letting the user in. Encryption can be run by the AES and SSL existing standards which are widely known and used on banking websites. The company website should be configured to only allow browsers using **http*s*** rather than plain http.

Without a comprehensive security policy, including an operating system that has security features in use, important private data both about the business as well as customers can be compromised by a malicious person, and the customers will lose trust in the company. Also, there exist laws in many countries that require businesses to take computer security seriously and have policies enforced (Cussol, 2023).

***Device Drivers***

Device drivers are special types of programs that are designed to instruct the operating system on how to control a specific hardware component. They ‘relay requests for device access and actions from the operating system and its active applications to their respective hardware devices’ (Gillis & Tittel, 2022). Drivers may be built into the operating system’s files or supplied with new hardware that a person may purchase, depending on the component in question. Drivers allow applications to use different peripherals and accessories that didn’t come built-in to the computer and its operating system, opening up to a more user-friendly and immersive computing experience.

1 example of the benefit of drivers is if the company needed to provide alternative means for employees to work due to disabilities, they may need special hardware that allows for typing by voice, or magnifying text, or vibration motors for hearing-impaired people. Drivers make it possible for people who have disabilities to use computers too. Another example, one that has become popular during the COVID-19 pandemic, is video conferencing. Drivers are needed so the computer can communicate with the camera, speakers/headphones, and microphone so users can communicate via the Internet.

Not having drivers leads to a need for hardware engineers who will have to code from scratch the functionality of every single component the OS can not handle, which can lead to a large ‘reinventing the wheel' situation, as many common drivers ship with operating systems, and for others, they are typically supplied by the manufacturer of whatever component or accessory one purchases. Hiring these specialised hardware engineers will likely cost more than buying a more well-known operating system and computer off the shelf. It is impractical to have a computer that can not interact with any of its connected components, including the keyboard, mouse, and monitor - all of which need drivers, too (Gillis & Tittel, 2022).

***Fault Tolerance***

Fault tolerance is the ability for a computer system to continue running despite some components failing due to natural disaster, power outage, or hardware defect manifesting. Developing fault tolerance typically involves use of an uninterruptible power supply (UPS) unit (a type of device with a rechargeable battery that can keep computers powered for a number of hours during a power outage), RAID-1 disk mirroring, and frequent data backups with the data stored at another secure location (Kranz, 2023).

Fault tolerance in the form of UPS and RAID-1 will allow applications to continue running to complete their operations, possibly through the close of the business day, giving ample time for the IT team to procure new hardware (if a hard drive fails) or allow the power company to repair electrical systems (if electricity is out). Application data should still continue to be backed up with off-site backups in case of natural disaster or simple mistaken deletions.

Users will be able to save their work during a power outage without fear of losing a day or more of work and the company’s senior staff will be able to rest more easily knowing they did not lose any data during a power outage or due to a single hard drive failing. The company should still implement a backup plan, perhaps weekly overnight, of all computer data in such a way that restoration is also easy should someone accidentally delete something or a natural disaster takes out any physical structures. The weekly backups should be driven away from town to another location with secure storage protected from the elements.

Not having fault tolerance could mean a full business failure or major setback should the computer system(s) fail without backups. Not having a UPS unit will cause the system to simply not run during power outages. Not having RAID will mean there would be no way to have data synced up and mirrored across multiple hard disks, meaning all mirroring will need to be manually carried out by the IT team via manual copying of drives, with system downtime during these processes and the business being unable to function during these periods. Not having data backed up and stored at another secure location could result in business failure and the inability to re-open should the primary location be damaged or lost in a natural disaster. Ultimately, all of these results from a lack of fault tolerance have extremely impacts and are penny wise but pound foolish.

## Function, Management, and Maintenance Requirements

The organizational requirements from TSI’s back-office include the ability to run more than one program at once; full office suites; the ability to connect computers to office peripherals like printers, scanners, and fax machines; and Internet connectivity (Gallaher, 2024). From the perspective of operating system support, these abilities would translate to multiprogramming, multiprocessing, multithreading, virtual memory, and device drivers (Novasak, 2024).

The security requirements are user authentication, specific user permissions (known as access control), according to job position to change files/perform other computing tasks, and protection from threats on the network/Internet (Gallaher, 2024). For the operating system, these requirements would be handled by usernames and strong passwords, encryption for protection from unauthorised users accessing data, a firewall for network security, antivirus/anti-malware software to ‘protect the system from viruses, malware, and other malicious software’, and updating the operating system patches so security vulnerabilities aren’t taken advantages of by cyber criminals (GeeksForGeeks, 2023a). Laws in various countries also require businesses to have security procedures for any data on their citizens a business may store (GeeksForGeeks, 2023a).

Some performance problems TSI has been experiencing with their existing TSI OS that were indicated were the inability to run more than a single program at a time, the system locking up when an interrupt is generated by a second processor due to lack of support for more than one processor, and limited system call support commands of only ‘basic file open, close, read, and write’ (Gallaher, 2024). Programs that require multiple threads hang on launch (Gallaher, 2024). TSI OS requires frequent reboots due to it running out of memory (virtual memory support would solve this problem) (Gallaher, 2024).

Reliability issues are present with TSI OS that could be solved with a general purpose operating system (Gallaher, 2024). The first has to do with TSI OS supporting only single-core CPUs, which aren’t being made anymore and as such TSI needs to buy older processors that are no longer supported by their manufacturers (Gallaher, 2024). The lack of an adequate system call interface also causes compatibility issues with software that would be essential in any office setting, such as a word processor, spreadsheet, calendar, and email (Gallaher, 2024). System calls are how the operating system ensures the correct user and process are accessing the desired devices or files during a computing session (JavaTpoint, n.d. b). The calls are for process control, file usage, device management, essential statistics about processes, communication across the system, and networking (including getting the computer online and sending/receiving data packets) (JavaTpoint, n.d. b). Fault tolerance is a big thing that doesn’t exist in TSI OS that is essential in any business, large or small (Gallaher, 2024). This is linked to a lack of device drivers for TSI OS in that since no drivers exist for TSI OS to operate backup devices like RAID cards, tape drives, optical drives, or external hard drives, there is no way for TSI OS to automatically backup the data on the hard drives, resulting in TSI storing all of their data on separated individual hard drives, possibly using other computers to backup the data stored within (Gallaher, 2024). Backup is also important for maintenance of the computer system as it is important to both be able to store and find records across the duration of a business’s existence.

**Computer Architecture**

This next section is to cover what kind of computing hardware will be needed to host a general purpose operating system. Red Hat Enterprise, a version of the Linux OS with up to 5 years’ full technical support and another 5 years’ maintenance support (Red Hat, n.d. a), will be the OS being considered.

**Required Hardware to Host the Operating System**

According to Red Hat in their Customer Portal - *9.3 Release Notes - Chapter 2. Architectures*, the requirements for the current version, 9.3, are any of the following computer architectures, with the minimum variant of each architecture enclosed in curly bracket symbols ( ) (Red Hat, n.d. b):

* AMD and Intel 64-bit architectures (x86-64-v2);
* The 64-bit ARM architecture (ARMv8.0-A);
* IBM Power Systems, Little Endian (POWER9);
* 64-bit IBM Z (z14).

According to Joel Arellano, of Cybernet Blog, the Intel 64-bit architecture (shorthand written as ‘x86-64’) is the most common kind of CPU in use in businesses, with 71% market share as of 27 June 2023 (Arellano, 2023). This will be the architecture being analysed going forwards. All Intel CPUs mentioned by Arellano, from the i3 up to the i9, are multicore with a range of 4 to 24 cores on 1 chip, allowing different numbers of threads per process or multiple processes to run (Arellano, 2023).

Other hardware requirements for Red Hat Enterprise Linux, as stated in their *Red Hat Enterprise Linux Technology Capabilities and Limits* web page from 1 November 2023, include at least ‘1.5 GiB for local media or NFS network installation, 3 GiB for HTTP(S) and FTP network installation’ for the minimum quantity of RAM and ‘10GB minimum, 20GB recommended’ for minimum hard disk / solid state drive free space (Red Hat, 2023). Red Hat’s *Appendix A. System requirements reference* also mentions a video requirement of at least 800 x 600 and 1026 x 768 monitor resolution (Red Hat, n.d. c). All of these requirements are minimums and one should expect to purchase hardware that has capability to do more and have room for future capabilities. For instance, the author’s computer has an Intel i9 CPU, 32 GiB RAM, 1 TB solid state drive, and a 34-inch (86 cm) monitor capable of a resolution of 3440 x 1440, which cost approximately $1400 to build (with $450 on the monitor) in late 2020.

**Support and Functionality of the Architecture**

According to Satish Kumar’s article entitled *Process Memory Management in Linux* from TutorialsPoint, memory management is done via static and dynamic allocation (Kumar, 2023). Static memory allocation is where the memory is allocated when the program is compiled (Kumar, 2023). The allocation done in this manner can not be changed afterwards (Kumar, 2023). Dynamic allocation is where memory can be allocated and such allocation can be adjusted while a program is running via ‘system calls such as malloc(), calloc(), and realloc() to dynamically allocate memory’ (Kumar, 2023). Virtual memory is another memory management technique Linux uses that has the kernel's Virtual Memory Manager (VMM) swap data out of RAM and into space that is on the hard disk using the ‘mmap() system call’, which ‘maps a file into a process's virtual memory address space, allowing [the] process to read and write to file's contents as if it were part of its own memory’ (Kumar, 2023). Use of virtual memory is also known as swapping, and especially happens when RAM availability is low (Kumar, 2023). Shared memory is something else Linux is capable of, which is where multiple processes can access the same area of memory via 3 more system calls: shmget(), which ‘creates a shared memory segment’, shmat(), which ‘attaches [a] shared memory segment to a process's address space’, and shmdt(), which ‘detaches [a] shared memory segment from process's address space’ (Kumar, 2023). The Linux kernel has its own memory management techniques, namely to deal with memory fragmentation, memory leaks which is ‘where memory is not returned to system and can eventually cause program to crash due to insufficient memory’, and memory protection, which is where the memory management unit utilises user permissions for each memory pages to prevent access by unauthorised users (Kumar, 2023).

According to Surajit Saha, of Scaler Topics,, process management in Linux is done by way of commands such as the 11 that follow (Saha, 2023):

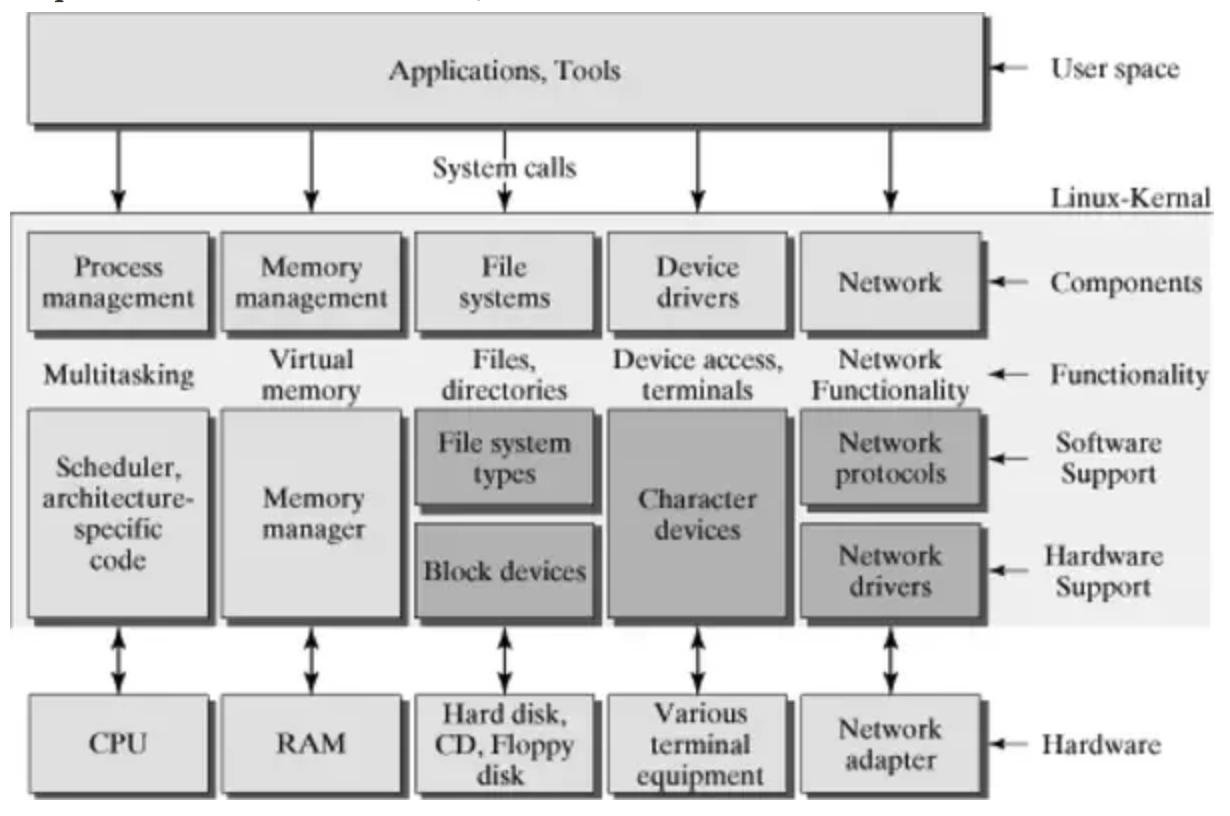
* ps, which displays the currently running processes and information about them;
* top, which displays resource usage about system processes in real-time;
* kill, which terminates a process;
* nice (and renice), both of which change the priority of a process that is currently running;
* ps PID, which shows the state of a specific process;
* pidof, which shows the process ID of a specific process;
* df, which shows the system’s disk management (hard drive partitions, how much space is used and free, their names, and where they are mounted in the filesystem);
* free, which shows the RAM’s status (how much is used, free, shared, and available for both physical and virtual memory;
* bg and fg, which send a process to the background or foreground, respectively.

In addition, processes are broken up into foreground and background processes, which also have user designators of System and User (Saha, 2023). Foreground processes are the ones that the user is interacting with, such as when writing an email or playing a game (Saha, 2023). Background processes are processes that the user doesn’t interact with and just run silently, like a process that may periodically check for new emails, an antivirus scanner app, and a minimised web browser window (Saha, 2023). System processes are initiated and controlled by the kernel and User processes are initiated and controlled by the computer’s user (Saha, 2023). Processes can have any of 5 states: running, sleeping, stopped, zombie, and orphan (Saha, 2023). A process that is running is executing in the CPU (Saha, 2023). A stopped process is one the user terminated (Saha, 2023). A zombie process is one that has finished executing but hasn’t been cleaned out from the system yet (Saha, 2023). An orphan process is one where its parent has been terminated. A sleeping process is one that is waiting for an available resource (Saha, 2023).

According to Prajwal Gurumurthy in an article entitled *Understanding IO in detail [part 1]!*, I/O, or input/output, is how data moves around a computer system and its components, including how the data gets transferred to external devices and the Internet (Gurumurthy, 2019). In Linux, all I/O units, including physical hardware like the monitor used to display content, are abstracted as files (Gurumurthy, 2019). Figure 1, shown below, shows some of these devices and how they interact with each other in a computer system.

**Figure 1**

I/O Devices



*Note*: Picture is COPYRIGHT Prajwal Gurumurthy (2019).

Other devices used include the northbridge and southbridge interfaces, which handle communication of components around the motherboard, various buses which are how peripherals and external devices are connected to the motherboard, which include ISA, PCI, USB, SCSI, and VESA (Gurumurthy, 2019).

According to Tyler Carrigan, from Red Hat’s editorial team, a tool called iostat, which is part of the sysstat package, can be installed and used to find out if any devices are transferring data at slower than normal speeds (Carrigan, 2020a). One runs it by opening a Terminal window (command line) and executing the command iostat (Carrigan, 2020a). If it’s not installed, the user will need to install the sysstat package first (Carrigan, 2020a). After it’s installed, running it will display transfers per second, KB written, KB read, total KB read, and total KB written (Carrigan, 2020a). These units can be displayed in MB by adding the option -m to the iostat command (Carrigan, 2020a).

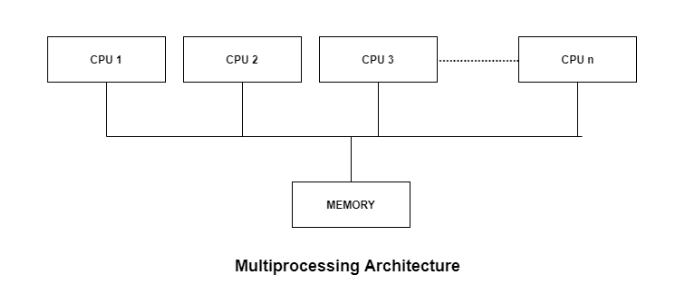
According to Red Hat in their Customer Portal - *Chapter 1. Overview of available storage options*, the Red Hat Linux supports both local storage on the computer itself using file systems XFS, Ext4, and Stratis; as well as remote storage via iSCSI, fibre channel (FC), non-volatile memory express (NVMe), device mapper (DM) multipath, and network file system (NFS) (Red Hat, n.d. d). RAID is also supported in Red Hat, according to their Customer Portal - *Chapter 19. Managing RAID Red Hat Enterprise Linux 9* (Red Hat, n.d. e). It supports firmware, hardware, and software RAID, with software RAID being performed via the mdraid subsystem which, in conjunction with external metadata, can ‘access Intel Rapid Storage (ISW) or Intel Matrix Storage Manager (IMSM) sets and Storage Networking Industry Association (SNIA) Disk Drive Format (DDF)’ (Red Hat, n.d. e). The mdraid subsystem is controlled with a utility called mdadm (Red Hat, n.d. e).

**Architectural Issues in Multiprocessor Systems**

Some issues multiprocessing computer systems can experience include synchronisation difficulties with ensuring processes are executed in the correct sequence, higher power consumption, and limited performance gains especially if a given program wasn’t developed with more than one processor in mind (GeeksForGeeks, 2023b). According to Kristi Castro, of Tutorials Point, multiprocessor systems share the same memory, bus, peripherals, and other devices that connect to the motherboard, and come in 2 flavours: asymmetrical and symmetrical (Castro, 2023). Symmetric multiprocessing is where all processors share the operating system and communicate with each other as peers (Castro, 2023). In asymmetric multiprocessing, one of the processors acts as a ‘master’ unit that tells all the others the system what to do (Castro, 2023). Castro also states that multiprocessor systems may be cheaper over the long run as they are faster but they are still more expensive at the outset (Castro, 2023). They also require a more complicated operating system to schedule processes and handle resources spread between the processors vs having only 1 processor (Castro, 2023). Finally, more RAM is needed to handle multiple processors (Castro, 2023). All of these aspects put together especially in light of the possibility of diminished performance returns pointed out by GeeksForGeeks (2023b) depending on whether the programs one will use on the computer(s) may not make going with a multiprocessor system logical at an office, where 1 processor that has multiple cores can perform just as well for the tasks that need to be done and at a much more economical price. Figure 2, shown below, is Castro’s illustration of the sharing of multiple processors and the memory of a computer system.

**Figure 2**

Multiple CPUs sharing memory



*Note*: Image is COPYRIGHT Kristi Castro (2023).

**Process Management**

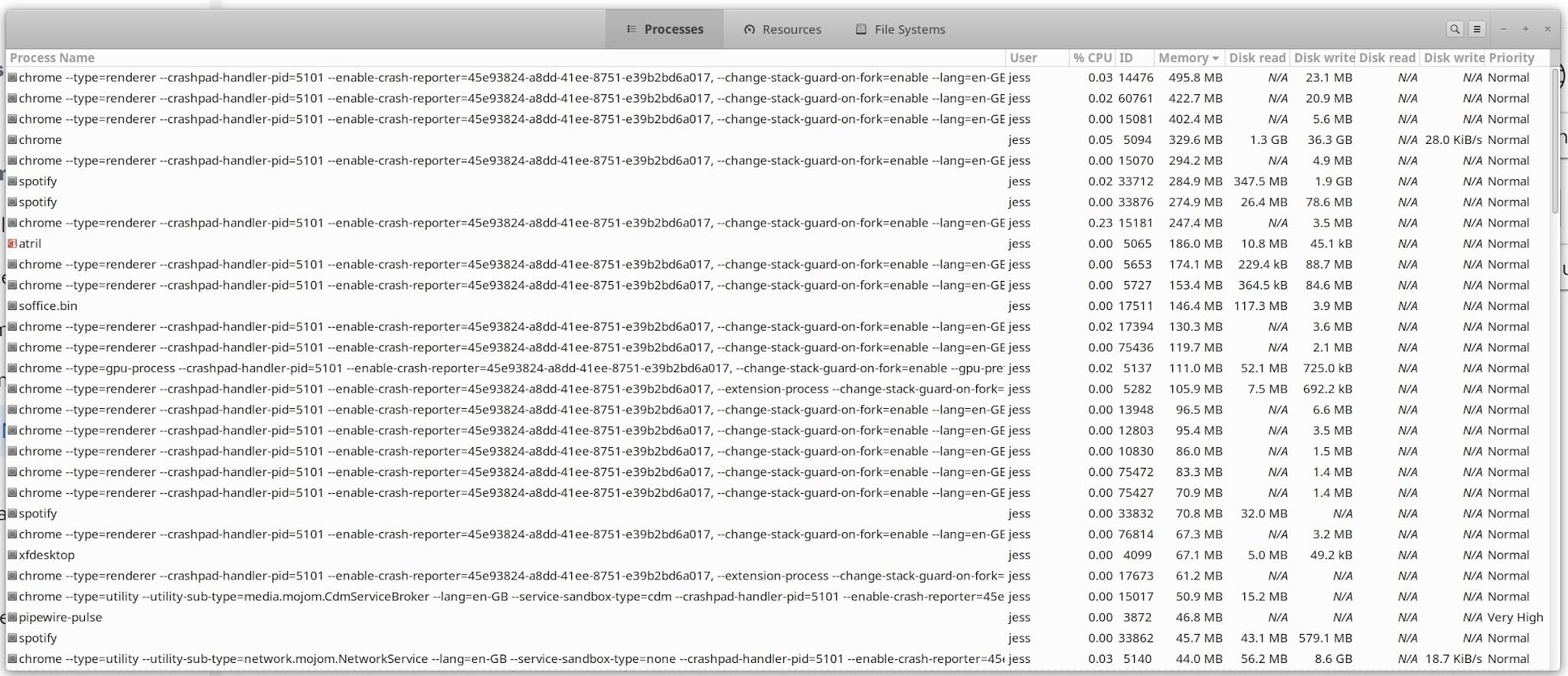
This next section is for covering how Linux (including Red Hat Enterprise) manage processes in the OS.

**Responsiveness to Organizational Requirements**

According to Red Hat in their website Customer Portal - *Chapter 24. System Monitoring Tools Red Hat Enterprise Linux 6*, one can use the commands ps aux and top, as well as the System Monitor Tool to view running processes and how much memory and processing cycles they are using (Red Hat, n.d. f). The System Monitor Tool is not installed by default, but one can install it from a Terminal window by running the command yum install gnome-system-monitor as the root user (Red Hat, n.d. f). One can run the System Monitor Tools by ‘either select Applications → System Tools → System Monitor from the panel, or type gnome-system-monitor at a shell prompt’ (Red Hat, n.d. f). A screenshot of the author’s instance of System Monitor is shown in Figure 3, where the computer is running LibreOffice, Spotify, Atril Document Viewer, Thunar File Manager, and several tabs of a Chrome browser.

**Figure 3**

The author’s computer is displaying System Monitor as of 2024-04-27 17:59 UTC. The columns can be clicked to sort on the desired category. This instance is sorted based on memory usage.



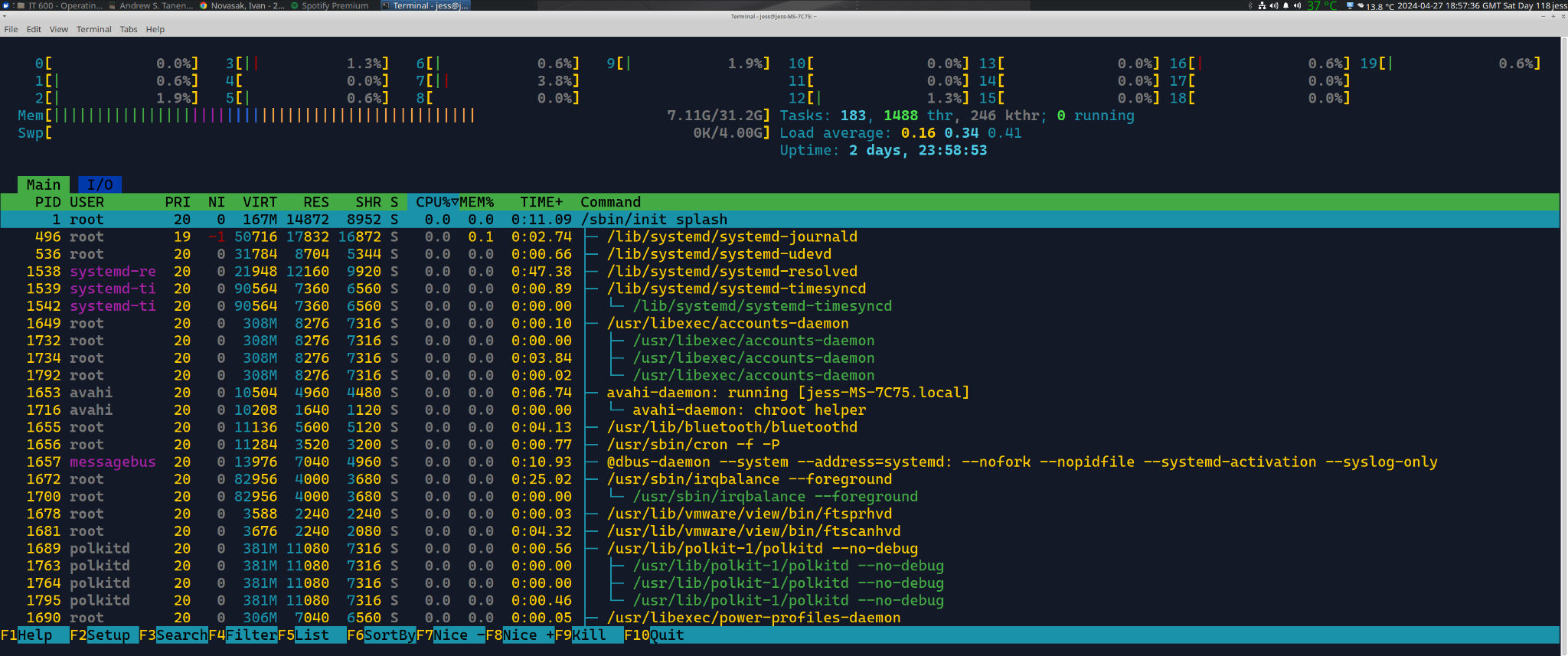
Right-clicking on a process shows options to display Properties about the process, show Memory Maps, display a list of files the process has open, change priority, set affinity, stop, continue, terminate, or kill the process. The ability to sort these processes on categories like memory usage, CPU utilisation, and disk read/write usage are quite useful as one can see at a glance if a given process or set of processes is using a lot of resources. In the upper-right corner of the System Monitor tool window, there exist icons to search for a specific process as well as view active processes, only the current user’s processes, processes opened by every user on the system, toggle to show a process’s dependencies, a Preferences menu to toggle more process data in System Monitor, as well as a Help section.

**Software Tools for Thread Analysis and Deadlock Detection**

For thread analysis in Linux, Stephen Rauch mentions a tool known as htop can be used to check which processes’ threads are running and which process they are associated with. This tool can be installed on Red Hat Linux using a similar command to how one installed in the Terminal window via the command yum install htop as the root user (Gite, 2024). When one runs htop in the Terminal window after installation, tree view can be toggled by pressing F5 on the keyboard. Figure 4 shows the author’s computer’s output of this program as a screenshot.

**Figure 4**

Screenshot showing htop tree view for showing operating threads



According to Amitesh Anand, of LinkedIn and writing in a CodeCraft Chronicles issue, a deadlock is when, ‘two or more processes are unable to proceed because each is waiting for the other to release a resource. This can result in a complete system halt and adversely affect the overall performance of your operating system’ (Anand, 2023). For detection and handling of them, Anand states that Linux handles deadlocks via 2 different methods:

1. The Ostrich Algorithm, which is where the operating system assumes they just won’t happen and that they are exceptionally rare, instead focusing on other ways to enhance performance (Anand, 2023). This method doesn’t address deadlocks head-on, but the algorithm puts emphasis on ‘avoiding resource contention issues’ by building a powerful system that can handle many obligations with ease (Anand, 2023).
2. Deadlocks can be detected in Linux via a program called the Out-of-Memory (OOM) Killer, which just terminates the processes that are causing the deadlock, which will allow the system to continue functioning amongst the rest of the processes (Anand, 2023).

Anand also recommends 3 ways software developers can avoid deadlocks happening by (a) implementing a timeout mechanism that will take alternative action if a process is waiting too long for an event to occur, (b) ‘analyze system logs and performance metrics to proactively detect and resolve deadlock issues before they impact critical operations’, and (c) allocate resources in the correct sequence and avoid having unnecessary dependencies (Anand, 2023).

Tarun Sharma, a Software Engineer at Cisco, recommends the following steps for trying to detect deadlocks in Linux:

1. ‘Run strace -p <pid> and check if the process is stuck in wait.’
2. ‘If you suspect your application is \*stuck\*, do a ps aux|grep <application name>. If the output is "D" (uninterruptible sleep), it \*could\* mean there's a deadlock in your code.’
3. ‘Linux kernel has an inbuilt feature called lockdep. It can help in pin-pointing the line in code causing the deadlock.’

**Multiprocessing Support**

According to Red Hat, Red Hat Enterprise Linux 9 supports up to 8192 logical CPUs, with 1792 having been tested (Red Hat, 2023). Regarding what a logical CPU is in this context, Red Hat says ‘every core/thread in a multicore/thread processor is a logical CPU’ (Red Hat, 2023). E.g. the author’s Intel i5 CPU has 10 cores, each having 2 threads, so in Red Hat’s nomenclature that would sum to 20 logical CPUs. Red Hat Enterprise Linux supports both Non-Uniform Memory Access (NUMA) and Symmetric Multi-Processor (SMP) as such to keep CPU cores busy and by doing load-balancing of threads from each process around the cores (Red Hat, n.d. g). Red Hat Enterprise Linux schedules processes based on 2 categories: realtime policies and normal policies (Red Hat, n.d. h). Realtime threads are top priority and as such will run before any normal threads (Red Hat, n.d. h). Threads amongst each of these categories are scheduled according to either first in first out (FIFO) or round-robin (RR) where time is sliced up such that all processes get an equal share (Red Hat, n.d. h).

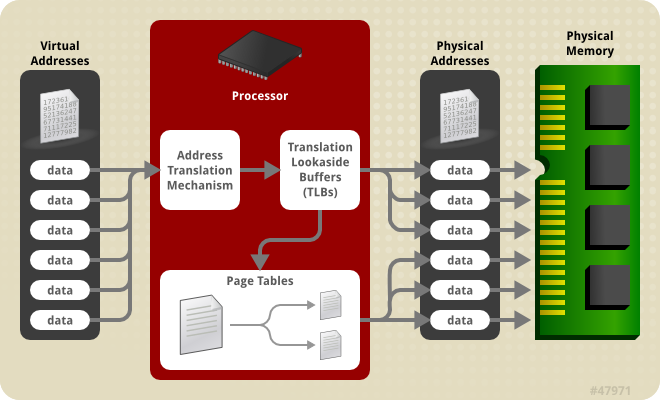
**Memory Management**

**Memory Abstraction**

According to Red Hat in *Red Hat Enterprise Linux 7 Reference Guide - Chapter 2. Memory Allocation*, Linux uses a virtual memory system to represent RAM in the OS and uses page tables and translation lookaside buffers (TLBs) which are components in the processor that link virtual memory addresses to physical memory addresses in an address translation mechanism (Red Hat, n.d. i). Figure 5 shows this linking in action, with the virtual addresses on the left side of the image, the processor’s relevant components in a red box in the centre, and the physical addresses as well as the RAM chips on the right.

**Figure 5**

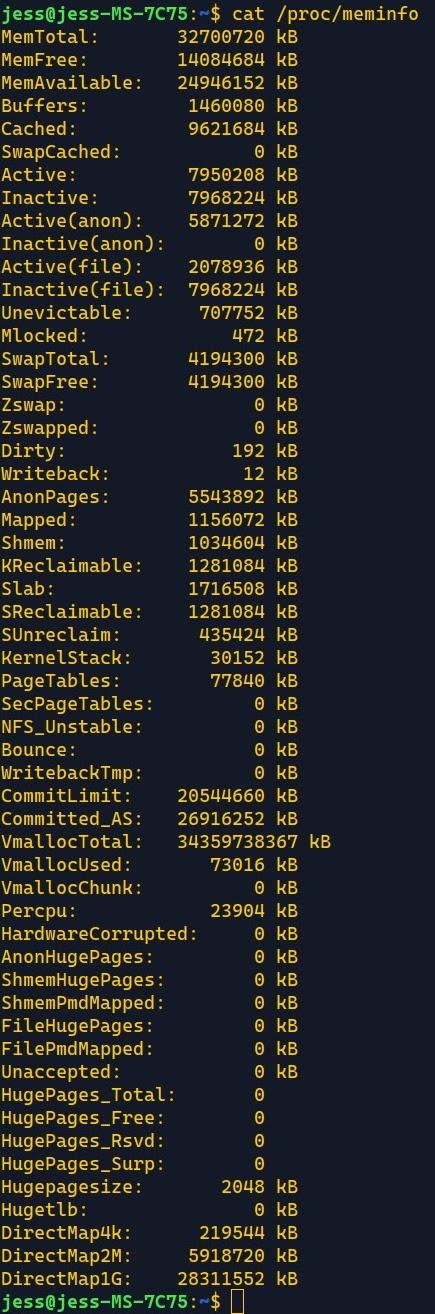
‘Red Hat Enterprise Linux for Real Time Virtual Memory System’



*Note*: Image is COPYRIGHT Red Hat (Red Hat, n.d. i).

Narendra, of RedSwitches, recommends using the cat /proc/meminfo command which will display statistics about the memory in use from a file in the Linux system (Narendra, 2023). The units are displayed in kilobytes, so one will need to divide by 1024 to get a value in MB and again by 1024 to get it in GB (Narendra, 2023). The information displayed is ‘real-time data on the system’s memory utilization and the kernel’s use of shared memory and buffers’ (Narendra, 2023). Figure 6 shows the author’s computer’s memory usage as displayed by this utility on 2024-04-27 22:02 UTC.

**Figure 6**

Memory usage statistics output from cat /proc/meminfo

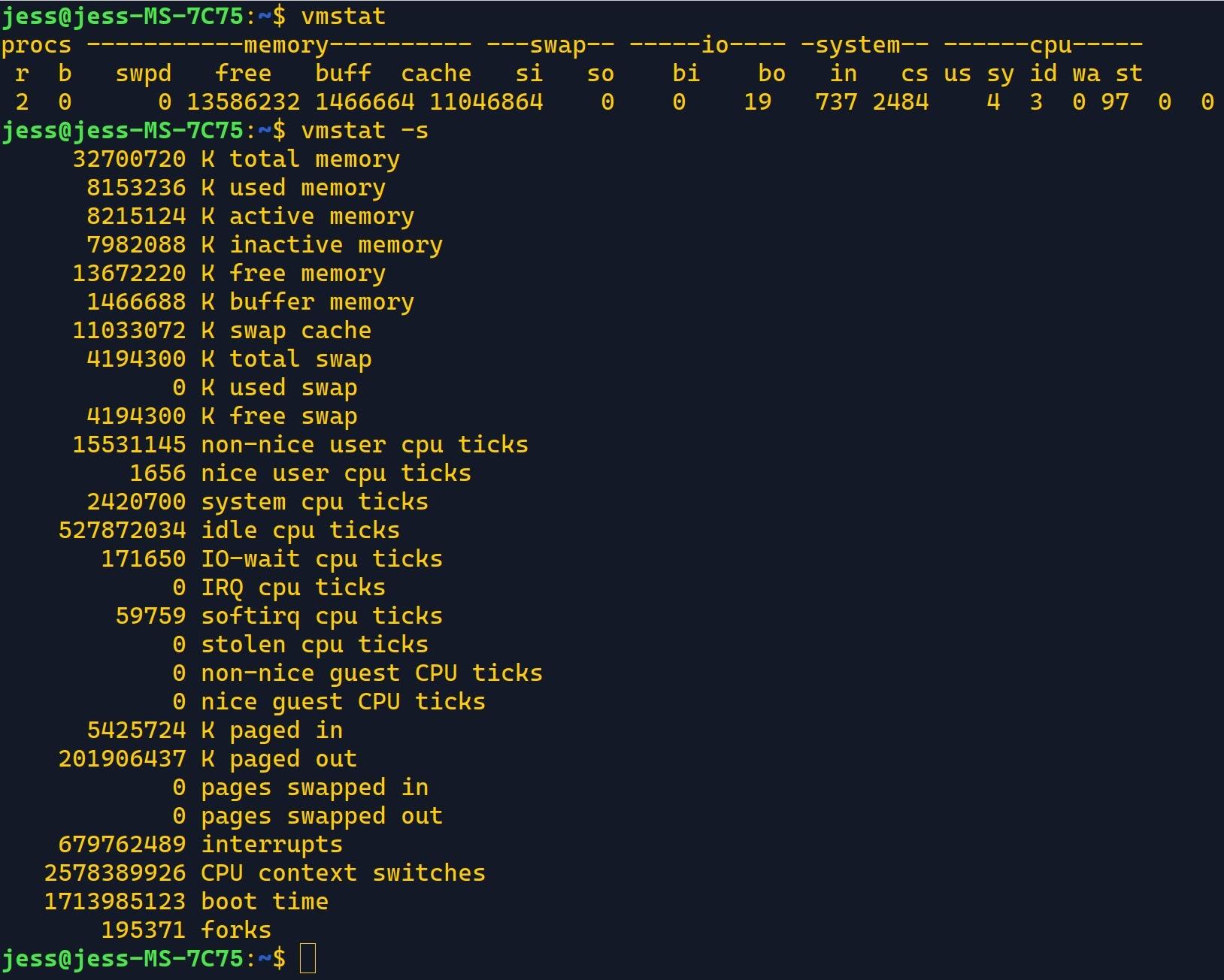
**System Support for Virtual Memory, Memory Paging, and Segmentation**

Surendranath P, Senior Technical Leader at KPIT and writing for LinkedIn, says Linux handles memory segmentation with 2 approaches: software based and hardware based (P, 2024). He says that with hardware based segmentation, each logical segment of memory is stored as a base and limit value in the segmentation table (P, 2024). The difference between software and hardware segmentation, according to Surendranath P, is: ‘The difference between the software based and the hardware based segmentation is that in the software based segmentation, we use the final binary to store the segmentation table but in the hardware based segmentation, the CPU registers store the starting address of the segmentation table and eventually the segment numbers and the offsets.’ (P, 2024). Linux terminates processes that experience a segmentation fault, which is where a process tries to access a memory address that doesn’t exist (P, 2024).

According to Harsh Mange, of Bit Fetch, Linux uses a paging system known as demand paging, which is where the operating system uses fixed-size blocks of memory, known as pages, to reserve space for data to be stored (Mange, 2023). This is also known as virtual memory, because some space on the hard disk can also be used as an extension to the physical memory present in the system (Mange, 2023). According to Tyler Carrigan, of the Editorial Team at Red Hat, virtual memory statistics can be shown using the vmstat utility, with the argument -s being added to the command to show more statistics about memory usage along with CPU and I/O event counters (Carrigan, 2020b). The data on ‘swap’ represents the area of virtual memory that is on the solid state drive rather than in physical RAM (Carrigan, 2020b). Figure 7 shows the output of both the basic vmstat command as well as the vmstat -s command to show these memory statistics.

**Figure 7**

Outputs from vmstat and vmstat -s from the author’s computer’s Linux Terminal window



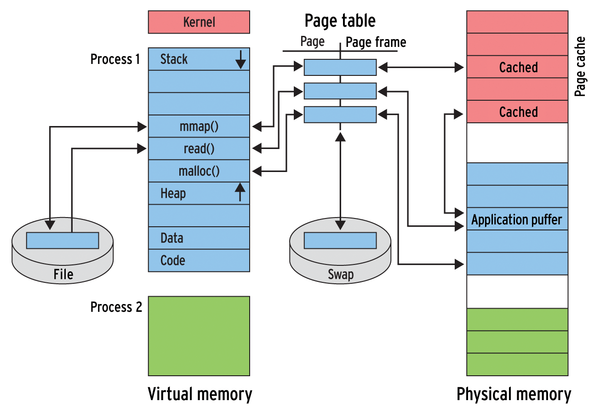
*Note*: The author’s computer can not be placed under heavy load at thai time due to overheating. The observed values are under normal usage.

**Techniques Used for Memory Management Policy and Mechanism Separation**

Memory management in Linux consists of 3 aspects: the memory management unit, the page fault handler, and the swap manager (Mange, 2023). The memory manager is responsible for allocating memory pages to applications that request them, deallocate them when a program is finished with the pages, and tracks how much physical memory (RAM) is being used and how much is free to be allocated (Mange, 2023). The page fault handler deals with the situation of when a program tries to access a page that isn’t in RAM - it instructs the OS to search the hard disk drive (or solid state drive) for the data to copy it into RAM (Mange, 2023). The swap manager is in charge of transferring the least frequently used data pages that are in RAM over to a reserved spot on the hard disk drive (or solid state drive) if the RAM is close to filling up and more is needed for a program to use (Mange, 2023). The swap manager can also transfer this data back into RAM should an application request it. David Both, a correspondent writing for OpenSource.com, says that ‘most Linux installations create a swap partition’ on the hard disk drive as a default, and that the size of this swap partition used to be twice the available RAM but beyond 2 GB of RAM it makes sense to use a smaller swap partition to reduce the chances of ‘thrashing’ (the OS repeatedly transferring data from RAM to disk and vice-versa, which is something that immensely slows down computers) (Both, 2020). He states that Fedora’s documentation recommends using 0.5 to 1.5 times the RAM if the computer has between 8 and 64 GB of RAM (Both, 2020). Alexander Hass and Willi Nüßer, writing for ADMIN Magazine in 2014, present an excellent image that shows a memory allocation and mapping system in action in Linux. It is shown in Figure 8 (Hass & Nüßer, 2014).

**Figure 8**

Memory management in Linux



*Note*: Image is COPYRIGHT Alexander Hass & Willi Nüßer (2014).

**I/O and Mass Storage**

This section discusses Input/Output, storage, and file systems.

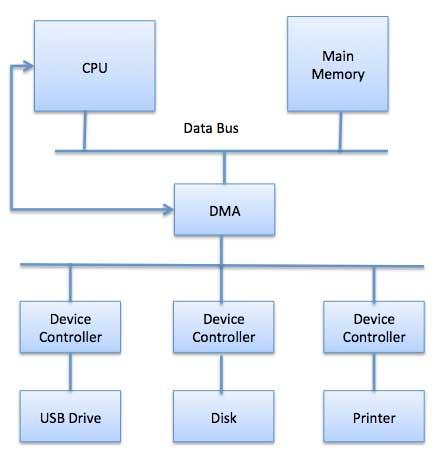
**Hardware-Software Interface for I/O Management**

Drivers are essential in a computer being able to communicate with hardware, including input/output (I/O) devices. According to Pallavi Godse, a SEO and PPC specialist writing for LinkedIn, drivers in Linux are handled differently vs in Microsoft Windows (Godse, 2018). In Windows, the manufacturer is responsible for making their hardware compatible with all current versions of Windows given that Windows is the most popular OS (Godse, 2018). With Windows, drivers have to be made specific to each versions separately - a hardware component that worked fine in Windows 8 or 10 is not guaranteed to work on 11 since drivers are tied to the OS version, which means every time a new version of Windows is released, there is a chance at least one peripheral a person invested in suddenly may not work due to no new Windows version drivers having been made for it (Godse, 2018). In Linux, drivers are handled at the kernel level and while typically it’s the open source community who ends up doing the most work to make drivers completely or improve drivers supplied by manufacturers, the drivers in Linux are typically forwards compatible with future versions of Linux (Godse, 2018). There are advantages and disadvantages to both approaches. If a peripheral or accessory is too new and happens to be released right before a new version of Windows is too, it may be possible that drivers won’t exist for either OS: the new Windows version as well as Linux. On the other hand, peripherals that are a bit older and may no longer have Windows support but still function fine may still be supported in Linux if drivers were made for it (Godse, 2018). Godse presents a case in his article where someone he knew had a device that had drivers only up to Windows 7 and so the peripheral didn’t work in Windows 10, yet it worked fine when connected to a laptop running Linux (Godse, 2018). Godse says this due to ‘the philosophy followed by the Linux kernel, and is that if a driver is added to a version of Kernel, all subsequent versions include it, and generally that driver is not removed until many years later, when the peripheral is not only obsolete but is not used by anyone. Generally, with the exception of the newer drivers we can be sure that our Linux systems will be 100% compatible with our peripherals, regardless of the years they carry the market and regardless of the type of operating system that is being used, provided that the operating system has a version of kernel supports’ (Godse, 2018).

According to Tutorials Point in their web page entitled *Operating System - I/O Hardware*, I/O devices are broken down into 2 major categories: block devices and character devices (Tutorials Point, n.d.). Block devices send and receive data as a stream of multiple bits at a time, whereas character devices send (and may receive, depending on the exact type of device) data one bit or character at a time (Tutorials Point, n.d.). Examples of block devices include hard drives, CD-ROM drives, media players, etc. Examples of character devices are keyboards, mice, serial ports, parallel ports, etc. (Tutorials Point, n.d.). Any I/O devices interface with the operating system’s drivers via a device controller, which ‘convert serial bit stream to block of bytes, perform error correction as necessary’ (Tutorials Point, n.d.). Typically communication with I/O devices still needs to go through the CPU but with direct memory access (DMA), the CPU can be bypassed for faster communication with devices that may be slow due to overhead from CPU interrupts (Tutorials Point, n.d.). Figure 9 shows this in action.

**Figure 9**

How I/O devices interact with a computer when DMA is involved



*NOTE*: Image is COPYRIGHT Tutorials Point India Private Limited (Tutorials Point, n.d.).

**File Systems**

For local storage, Red Hat Enterprise Linux 9 supports the XFS and ext4 file systems (FS), with XFS being the default for this OS currently (Red Hat, n.d. j). For storage on a network, NFS and SMB are supported, with SMB specifically for use on networks involving Microsoft Windows computers (Red Hat, n.d. j). Other FS include GFS2 and Stratis which are for larger computer clusters (Red Hat, n.d. j). Ext4 is backwards compatible with the prior versions ext2 and ext3, which may be useful if one has older hard drives with data on them stored in those file systems (Red Hat, n.d. j). According to Pure Storage, XFS has the ability to read and write data simultaneously, which is known as parallel I/O (Pure Storage, 2024). XFS was designed for reading and writing of large files, which makes this file system ideal for storing media files like music and video (Pure Storage, 2024). Parallel I/O is also good for database access as people can both request and return data simultaneously (i.e. if one person wanted to add data while another was requesting data to be read) (Pure Storage, 2024). Ext4 does not support parallel I/O, so simultaneously reading and writing can not happen under ext4, however it does support something XFS does not have: user permissions on individual directories, ensuring that users with specific roles can perform only designated operations in specific directories (Pure Storage, 2024). Both XFS and ext4 support journaling, which is a kind of meta data written to memory where in case power is lost or a drive crash happens, data being written isn’t lost and will be recovered upon reboot of the server (Pure Storage, 2024). XFS integrates backup and recovery into the file system, whereas ext4 does not, means on an ext4 system, backup will need to be performed manually (Pure Storage, 2024). Both of these file systems have their advantages and disadvantages, so it is best to use XFS for a server that will store large files and have access and write speed as the primary concern, whereas if file security amongst the users who will be accessing files is the bigger concern or if the computer will mainly be storing smaller files, ext4 is better (Pure Storage, 2024). If backwards compatibility with ext3 or ext2 is required, ext4 will also be a better choice as the ext4 driver can access those older file systems (Red Hat, n.d. j). The file system capacity is 1024 TiB for XFS and 50 TiB for ext4 (Red Hat, n.d. j).

**Relationship Between Scheduled Process Context Switching and I/O Interrupt Handling**

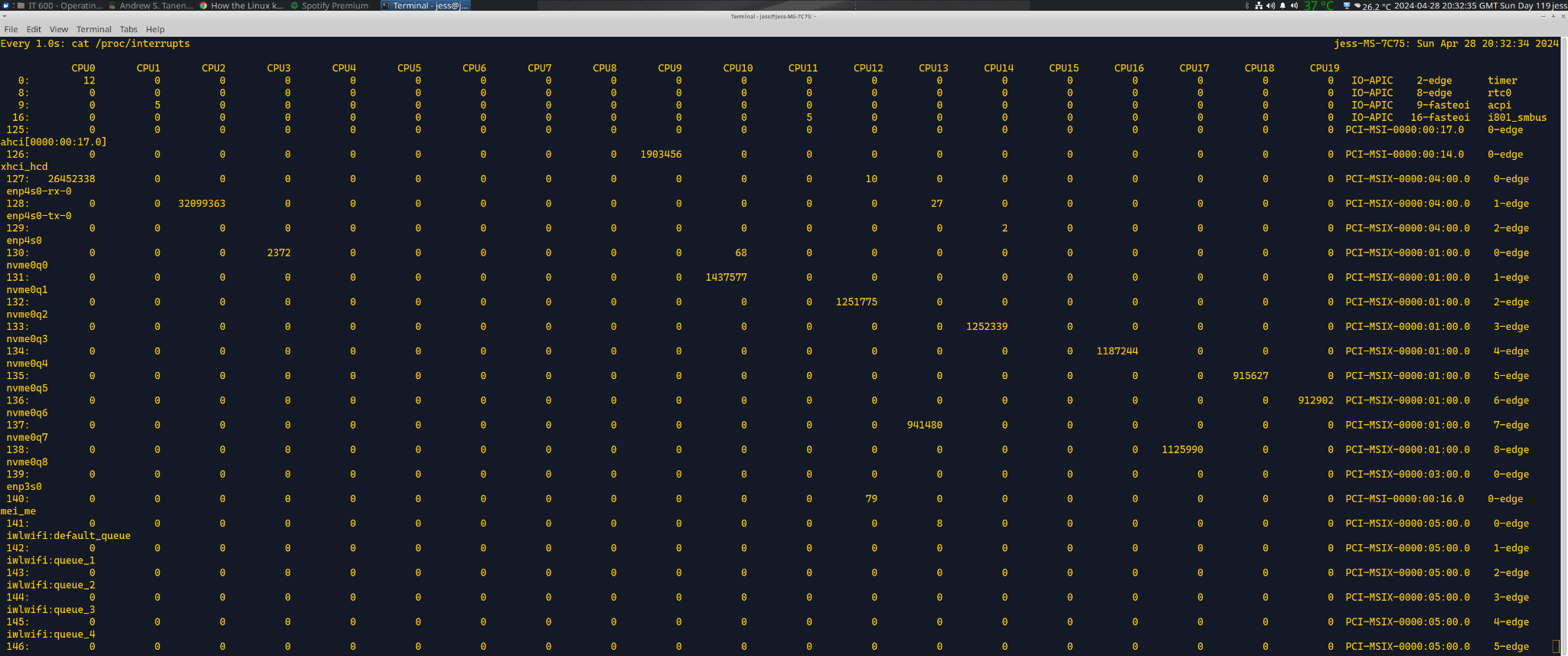
According to Stephan Avenwedde, a correspondent at OpenSource.com, an interrupt is a signal sent to the CPU that tells it to stop what it’s doing to switch to doing something else (known as context switching) (Avenwedde, 2020). There exist 3 types of interrupts: hardware, software, and exceptions (Avenwedde, 2020). An example of a hardware interrupt is when the user presses keys on their keyboard or moves the mouse - a hardware component is doing something that requires the CPU to switch context to follow the command that was just electrically sent (Avenwedde, 2020). A software interrupt is when a process instructs the CPU to do something, and can be time-invoked too, such AS when video and sound need to be synchronised like when watching a video or when data is being transferred to a device (Avenwedde, 2020). Exceptions are a type of interrupt that can be either from faulty commands or hardware failures (known as aborts) or if data needs to be transferred from RAM to the hard disk (page fault) (Avenwedde, 2020). Traps are another type of exception interrupt where a program can manually trigger the context switch, then afterwards the program can continue executing (Avenwedde, 2020). An example of a trap could be a program that is executing that then requires keyboard input, then resumes running after receiving entry. To view interrupts in Linux, one can run either the commands in a Terminal window (Avenwedde, 2020):

* cat /proc/interrupts
* watch -n1 "cat /proc/interrupts"

The latter command allows one to watch interrupts in real time (Avenwedde, 2020). Figure 10 shows the author’s interrupts from this command.

**Figure 10**

Terminal output of watch -n1 "cat /proc/interrupts" on the author’s computer



**Security**

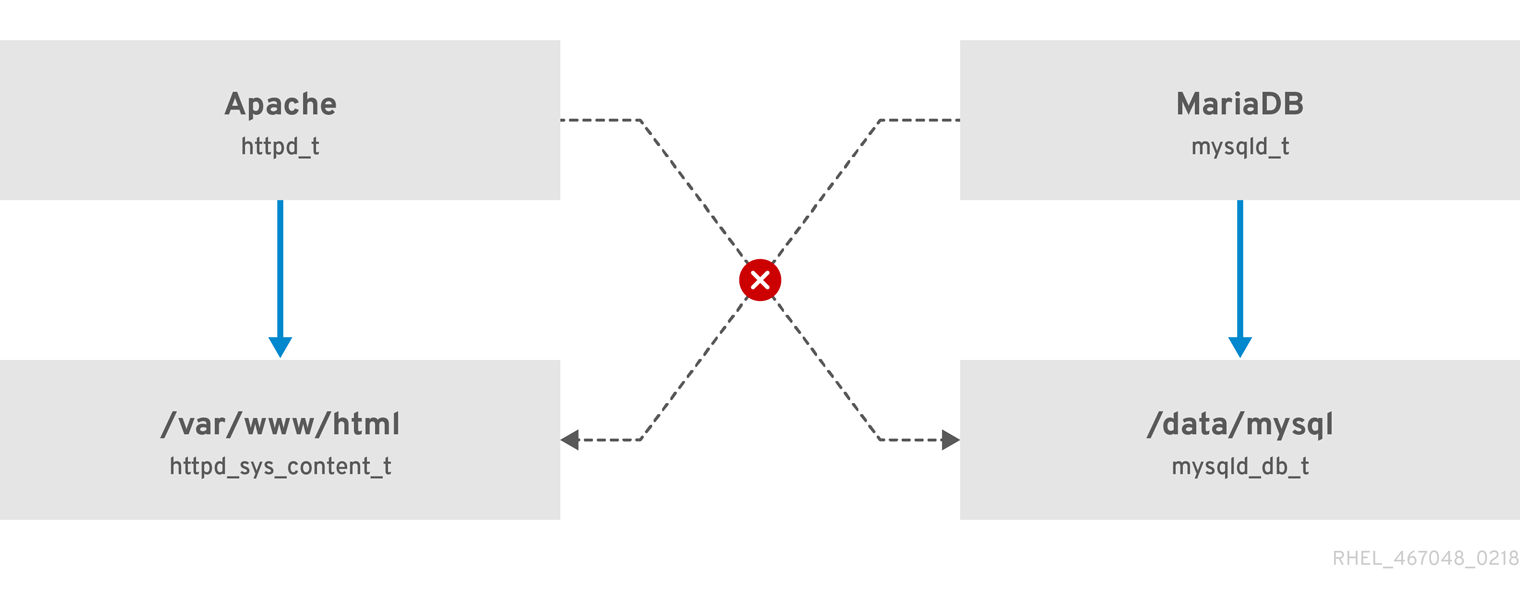
Next to be discussed is how security is handled in Red Hat Enterprise Linux.

**Security Model**

Security Enhanced Linux (SELinux), is a security system developed by a collaboration between Red Hat and the US National Security Agency (NSA) that makes the kernel act under the least-privilege model, known as Mandatory Access Control (MAC), where used and programs can do only what is explicitly defined by the system administrator (Culverhouse, 2017; Red Hat, n.d. l). The policies are enforced by context fields based on ‘user, role, type, and security level’ (Red Hat, n.d. l). One example of this in action is as follows as stated by Red Hat: ‘*There is a policy rule that permits Apache (the web server process running as httpd\_t) to access files and directories with a context normally found in /var/www/html/ and other web server directories (httpd\_sys\_content\_t). There is no allow rule in the policy for files normally found in /tmp and /var/tmp/, so access is not permitted. With SELinux, even if Apache is compromised, and a malicious script gains access, it is still not able to access the /tmp directory*’ (Red Hat, n.d. l). Figure 11 shows this graphically.

**Figure 11**

Diagram of LinuxSE enforcement



*NOTE*: Image is COPYRIGHT Red Hat (Red Hat, n.d. l).

Red Hat states that LinuxSE is not a replacement for antivirus software, passwords, firewalls, other security systems, or any all-in-one security system (Red Hat, n.d. l). It is still up to the system administrators to establish and enforce comprehensive cyber security policies in addition to this extra layer SELinux provides.

**Recommended Techniques and Best Practices**

Artem Karasev, a senior product marketing manager at TuxCare, has some recommendations for keeping a Linux system secure from malicious actors (Karasev, 2023):

1. Enable two-factor authentication (2FA) along with strong passwords that are unique to every user, specifically passwords with ‘at least ten characters, including special symbols and a mix of uppercase and lowercase letters’ (Karasev, 2023);
2. Using an SSH key pair for the server, which uses cryptographic key pairs and makes brute-force attacks much more difficult, which is worth the extra users who need to access the server will need to do given how much data can be there that is essential to any business (Karasev, 2023);
3. Keeping systems updated with the latest security patches, preferably automatically at regular intervals (Karasev, 2023). With Red Hat, one can get the latest updates by running the following command in the Terminal window: dnf update --security (Red Hat, n.d. k). To automate these updates in Red Hat, one cna follow the process listed in section 2.3, entitled *Installing security updates automatically*, located at <<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_and_monitoring_security_updates/installing-security-updates_managing-and-monitoring-security-updates>> (Red Hat, n.d. k).
4. Remove software that will not be used, including any third-party repositories (listings of online locations where software packages are stored) (Karasev, 2023). Unused software and their repositories can introduce vulnerabilities that may not have been a problem if that software were not present on production equipment (Karasev, 2023).
5. Disable the ‘root’ user and instead assign specific elevated permissions to a regular user that needs just enough access to maintain and upgrade the systems (Karasev, 2023).
6. Identify and close unused open ports on the network. Finding these ports can be done using a utility known as netstat (Karasev, 2023).
7. Enable a firewall like iptables to ensure one is not a victim of distributed denial of service (DDoS) attacks (Karasev, 2023). Firewalls can enable and deny traffic to and from specific IP addresses and are essential in keeping a server on a network secure from malicious activity (Karasev, 2023).
8. Conduct regular security audits and remain informed of the latest vectors of attacks so those can be dealt with before an attacker takes advantage of them (Karasev, 2023).
9. Use SELinux, which is Red Hat’s method of enforcing different security policies per user for the kernel (Karasev, 2023). This ensures only programs with the correct role can access system resources (Karasev, 2023).
10. Create and maintain backups of all data in case if one is victim of a security vulnerability, one can roll back to the previous backup after patching the vulnerability (Karasev, 2023). Also, test the backups to ensure they will work if they’re needed (Karasev, 2023).

**Overall Evaluation and Technology Recommendation**

The Linux OS comes in a variety of flavours, known as distributions. These can be overwhelming to someone new to the world of Linux, but there are ways to narrow things down based on what one is looking for, which Dave Taylor does in an article for Computer World magazine entitled *The 5 best Linux distros for the enterprise: Red Hat, Ubuntu, Linux Mint and more* (Taylor, 2018). All of the Terminal commands in this document will work on different Linux distributions, but the System Monitor graphical utility only works on distributions that have the GNOME desktop environment installed (it is not mandatory to be using GNOME to run such apps, though).

Red Hat comes with the GNOME environment by default. Taylor also mentions that Red Hat Enterprise Linux ‘includes integrated email, calendaring, contact management, a suite of office apps and virtualization capabilities to allow users to run Microsoft Windows and legacy apps as needed’. Red Hat Enterprise Linux Desktop’s pricing is $49 per seat and includes phone support, too (Taylor, 2018). Taylor also states that since Linux is open and flexible, ‘applications available on one Linux enterprise desktop platform is quite likely available on all the others too’ (Taylor, 2018). If a physical security key system as well as the ability to run Windows apps is needed at TSI, Red Hat Enterprise Linux natively supports Yubico’s key system and may be the way to go (Taylor, 2018).

Taylor then mentions another distribution, SUSE Linux Enterprise Desktop, that may be useful if TSI requires Microsoft Active Directory and Microsoft Exchange support (Taylor, 2018). The price of this distribution, which also includes phone support like Red Hat, is $120 per seat (Taylor, 2018).

Ubuntu is another Linux distribution that Taylor states is very popular and is translated into over 100 languages which may be ideal if TSI is planning to expand globally or employ people from widely disparate regions of the world (Taylor, 2018). For enterprise usage, Taylor has the following to say: ‘Ubuntu Desktop for Enterprise is supported through the Ubuntu Advantage program and costs $150/year with a minimum of 50 installs per organization’ (Taylor, 2018). Given the minimum of 50 installations and the $150/year subscription, Ubuntu Desktop for Enterprise may only be ideal if TSI were looking to both go global and have a sizable expansion.

It may also be helpful to contact each vendor and see if one can get a copy of each OS to run a pilot programme on some computers before making a full commitment to a specific distribution.

# References

Anand, A. (2023, May 23). *CodeCraft Chronicles: Deadlock Handling Strategies*. LinkedIn. <https://www.linkedin.com/pulse/deadlock-handling-strategies-amitesh-anand/>

Arellano, J. (2023, June 27). *Intel Processors: Which is Best Suited for Your Business Needs?* Cybernet. <https://www.cybernetman.com/blog/intel-processors-which-is-best-suited-for-your-business-needs/>

Avenwedde, S. (2020, October 5). *How the Linux kernel handles interrupts*. Opensource.com. <https://opensource.com/article/20/10/linux-kernel-interrupts>

Both, D. (2020, March 27). *An introduction to swap space on Linux systems*. Opensource.com. <https://opensource.com/article/18/9/swap-space-linux-systems>

Carrigan, T. (2020a, June 10). *Linux commands: exploring virtual memory with vmstat*. Enable Sysadmin. <https://www.redhat.com/sysadmin/linux-commands-vmstat>

Carrigan, T. (2020b, July 9). *I/O reporting from the Linux command line*. Red Hat. <https://www.redhat.com/sysadmin/io-reporting-linux>

Castro, K. (2023, September 14). *Multiprocessor systems*. Tutorials Point India Private Limited. <https://www.tutorialspoint.com/Multiprocessor-Systems>

Culverhouse, T. (2017, September 12). *SELinux (Security-Enhanced Linux)*. TechTarget. <https://www.techtarget.com/searchdatacenter/definition/SELinux-Security-Enhanced-Linux>

Cussol, E. (2023, September 1). *6 Reasons Why Data Privacy Is Important For Businesses*. Termly. <https://termly.io/resources/articles/why-is-data-privacy-important/>

Gallaher, M. (2024). *Milestone One Guidelines and Rubric - IT-600-Q3541 Operating Systems 24TW3*. Southern New Hampshire University. <https://learn.snhu.edu/d2l/le/content/1504944/viewContent/29875339/View>

GeeksforGeeks. (2023a, May 6). *Introduction of multiprocessor and multicomputer*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-of-multiprocessor-and-multicomputer/>

GeeksforGeeks. (2023b, June 19). *System protection in operating system*. GeeksforGeeks. [https://www.geeksforgeeks.org/system-protection-in-operating-system/](https://www.geeksforgeeks.org/system-protection-in-operating-system/#)

GeeksforGeeks. (2024, April 5). *Unix File System*. GeeksforGeeks. <https://www.geeksforgeeks.org/unix-file-system/>

Gillis, A. S., Peterson, S., & Lelii, S. (2021, July 12). *virtual memory*. TechTarget. <https://www.techtarget.com/searchstorage/definition/virtual-memory>

Gillis, A. S., & Tittel, E. (2022, December 9). *device driver*. TechTarget. <https://www.techtarget.com/searchenterprisedesktop/definition/device-driver>

Gite, V. (2024, March 22). *How to install htop on RHEL 9/8 using yum*. nixCraft. <https://www.cyberciti.biz/faq/how-to-install-htop-on-rhel-8-using-yum/>

Godse, P. (2018, May 3). *Linux Drivers: Advantages and Disadvantages at the same time*. LinkedIn. <https://www.linkedin.com/pulse/linux-drivers-advantages-disadvantages-same-time-pallavi-godse/>

Gurumurthy, P. (2019, March 1). *Understanding IO in detail [part 1]!* Github. <https://prajwalgurumurthy.github.io/2019/03/01/disk-io/>

Haas, A., Nüßer, W. (2014). *Managing Linux Memory.* ADMIN Magazine. <https://www.admin-magazine.com/layout/set/print/Articles/Managing-Linux-Memory>

JavaTpoint. (n.d. a). *Operating System Security*. www.javatpoint.com. <https://www.javatpoint.com/operating-system-security>

JavaTpoint. (n.d. b). *System Calls in Operating System (OS)*. www.javatpoint.com. <https://www.javatpoint.com/system-calls-in-operating-system>

Karasev, A. (2023, September 27). *Linux System Hardening: Top 10 security tips*. TuxCare/CloudLinux Inc. <https://tuxcare.com/blog/linux-system-hardening-top-10-security-tips/>

Kirvan, P. (2022, May 26). *multithreading*. TechTarget. <https://www.techtarget.com/whatis/definition/multithreading>

Kranz, G. (2023, November 14). *fault tolerance*. TechTarget. <https://www.techtarget.com/searchdisasterrecovery/definition/fault-tolerant>

Kumar, S. (2023, March 24). *Process Memory Management in Linux*. Tutorials Point India Private Limited. <https://www.tutorialspoint.com/process-memory-management-in-linux>

Mange, H. (2023, April 21). *How memory management works in Linux*. Bit Fetch. <https://harshmange.hashnode.dev/how-memory-management-works-in-linux>

Narendra. (2023, October 26). *How to Check Memory Usage in Linux – 5 Quick Ways*. RedSwitches Pte. Ltd. <https://www.redswitches.com/blog/check-linux-memory-usage/>

Novasak, I. (2024, March 17). *Top Secret, Inc: Analytic Organizational Profile*. Southern New Hampshire University.

P, S. (2024, January 20). *Memory management in Linux - Part 3*. LinkedIn <https://www.linkedin.com/pulse/memory-management-linux-part-3-surendranath-p-efy9c/>

Pure Storage. (2024, January 24). *XFS vs. Ext4: Which Linux File System Is Better?* Pure Storage, Inc. <https://blog.purestorage.com/purely-informational/xfs-vs-ext4-which-linux-file-system-is-better/>

Rauch, S. (2017, April 2). *Is there a way to see details of all the threads that a process has in Linux?* Unix & Linux Stack Exchange. <https://unix.stackexchange.com/questions/892/is-there-a-way-to-see-details-of-all-the-threads-that-a-process-has-in-linux/899#899>

Red Hat. (n.d. a). *Red Hat Enterprise Linux Life Cycle*. Red Hat, Inc. <https://access.redhat.com/support/policy/updates/errata?extIdCarryOver=true&sc_cid=701f2000001OH7EAAW>

Red Hat. (n.d. b). *Chapter 2. Architectures Red Hat Enterprise Linux 9*. Red Hat, Inc. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/9.3_release_notes/architectures>

Red Hat. (n.d. c). *Appendix A. System requirements reference Red Hat Enterprise Linux 9*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/performing_a_standard_rhel_9_installation/system-requirements-reference_installing-rhel>

Red Hat (n.d. d). *Chapter 1. Overview of available storage options Red Hat Enterprise Linux 9*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_storage_devices/overview-of-available-storage-options_managing-storage-devices>

Red Hat. (n.d. e) *Chapter 19. Managing RAID Red Hat Enterprise Linux 9*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_storage_devices/managing-raid_managing-storage-devices>

Red Hat. (n.d. f). *Chapter 24. System Monitoring Tools Red Hat Enterprise Linux 6*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/ch-system_monitoring_tools>

Red Hat. (n.d. g). *Chapter 4. CPU Red Hat Enterprise Linux 6*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/performance_tuning_guide/main-cpu#idm140718710538576>

Red Hat. (n.d. h). *4.2. CPU scheduling Red Hat Enterprise Linux 6*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/performance_tuning_guide/s-cpu-scheduler>

Red Hat. (n.d. i). *Chapter 2. Memory Allocation Red Hat Enterprise Linux for Real Time 7*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_for_real_time/7/html/reference_guide/chap-memory_allocation>

Red Hat (n.d. j). *Chapter 1. Overview of available file systems Red Hat Enterprise Linux 9*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_file_systems/overview-of-available-file-systems_managing-file-systems#local-file-systems_overview-of-available-file-systems>

Red Hat. (n.d. k). *Chapter 2. Installing security updates Red Hat Enterprise Linux 9 | Red Hat Customer Portal*. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_and_monitoring_security_updates/installing-security-updates_managing-and-monitoring-security-updates>

Red Hat. (n.d. l). *Chapter 1. Getting started with SELinux Red Hat Enterprise Linux 9.* Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/using_selinux/getting-started-with-selinux_using-selinux>

Red Hat. (2019, February 27). *What is the Linux kernel?* Red Hat. <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>

Red Hat. (2023, November 1). *Red Hat Enterprise Linux Technology Capabilities and Limits*. Red Hat Customer Portal. <https://access.redhat.com/articles/rhel-limits>

Saha, S. (2023, June 7). *What is Process Management in Linux?*. Scaler Topics. <https://www.scaler.com/topics/process-management-in-linux/>

Sharma, T. (2014, January 6). *Cmd Line Linux: Linux Kernel: Deadlocks and how to avoid them*. Blogspot. <https://cmdlinelinux.blogspot.com/2014/01/linux-kernel-deadlocks-and-how-to-avoid.html>

Sharma, T. (n.d.). *Tarun Sharma*. LinkedIn. <https://www.linkedin.com/in/tarun27sh/>

Sheldon, R. (2022, May 19). *multiprogramming*. TechTarget. <https://www.techtarget.com/whatis/definition/multiprogramming>

Taylor, D. (2018, January 9). *The 5 best Linux distros for the enterprise: Red Hat, Ubuntu, Linux Mint and more*. Computerworld. <https://www.computerworld.com/article/1715558/the-5-best-linux-distros-for-work-red-hat-suse-ubuntu-linux-mint-and-tens.html>

Tutorials Point. (n.d.). *Operating System - I/O hardware*. Tutorials Point India Private Limited. <https://www.tutorialspoint.com/operating_system/os_io_hardware.htm>

Williams, L. (2024, February 3). *System Call in OS (Operating System): What is, Types and Examples*. Guru99. <https://www.guru99.com/system-call-operating-system.html?gpp&gpp_sid>

Yasar, K. (2023, June 23). *multiprocessing*. TechTarget. <https://www.techtarget.com/searchdatacenter/definition/multiprocessing>