Comp 314: Graphic Design
ID: 3441988

Andrew Mackay

Operating systems assignment 4

1.

The difference between protection and security in operating systems mainly comes down to protection relating to the management of threats that are within the system to still maintain functionality like process, resources and control of access. Security is more so to defend the system from outside threats like to defend against external users trying to get in (malware, viruses, ransomware, phishing).(pedia, 2018)

Many families have a home computer with multiple family members that each have their own accounts, or an individual computer with maybe a guest account. This is protection as it protects one persons private information from other users accessing it by means of a password. There is also such thing as safeguards if there is a child user possibly you can safeguard their internet searches and restrict them from certain content, although this may not be entirely without fault.

On every computer there exists some file or information entered somewhere which can be used to either directly harm it's owner or gain inside knowledge on them so someone can further find evidence of manipulating or stealing from them. In 2021 tthere were 70,288 cyber crimes reported to the Canadian government, whereas just 4 years prior that number was at 27,829.(statcan.gc, 2022) This is a terrifying statistic that shows the steady growth in cyber crimes and gives insight into how crucial security is for entire companies, and individuals.

2.

The access matrix is a model of protection within an operating system that helps clearly define the rights of each process which is currently executing in the domain. It largely denotes when each process should begin each part of its own process, for if they were to all go at the same time the system would crash. This kind of protection is protecting internally against the system malfunctioning. (geeksforgeeks, 2019) Policy and mechanism in this situation may be because of the restrictions in place for some devices that maybe have within their policy they need to have priority at a certain point in a process whereas that isn't entirely reasonable or may be detrimental to the total set of processes running concurrently so the microkernal has to intervene to fix the path which is being set out according to policies in place.

3.

The hierarchical protection of the ring based model works by defining separate execution levels called privilege separation. Where the innermost ring provides the most set of privileges and the outermost the least. Code that runs during boot time has highest privilege and after boot, gets transported through something called a gate between levels of the hierarchy system. Within an interrupt the privilege system is also interrupted and code is transported to said highest level ring, which is restricted to a guarded code path. The highest ring can be thought of as the kernel mode ring.

Within the domain based protection model the software and hardware objects are treated as having individual properties which yield different levels of protection by the execution of the system. Like a case by case scenario, processes only have access to the objects in which they have authorization.

4.

It is important to separate processes of objects into specific domains as to ensure the objects process gets carried out effectively and minimizes interrupts and execution time, while prioritizing the processes of importance to go before lesser important ones. The confinement problem details the inability of the system to safeguard an object within its execution environment and is generally considered an unsolvable problem. We can attempt to control these processes somewhat successfully by means of the access matrix which categories which domain is allowed to have access to which process. This is directly related to the policies. When a process executes in user mode it can only invoke nonpriveledged instructions, and within its predefined memory space. These modes protect the OS while executing in kernal, from other processes executing in the user domain.

5.

Encryption is to take data which is likely sensitive in some way and to scramble it so it isn't translatable, whereas decryption is reversing the process. Authentication is the process of providing extra information to prove in a way you are who you say you are. Encryption based systems are much more secure but take are not as flexible as authentication based systems.(support.1password, 2022)

Although encryption is still used frequently in a variety of ways in modern times and encryption techniques also far predate the first computers. . Used for sending messages across a network, protect database data files, data and and disks from having their contents read unwillingly. There are also algorithms being part of the authentication branch of security which switch keys similarly to how we think of encryption algorithms.

References part A:

Silberschatz, A., Gagne, G., & Galvin, P. B. (2018). Operating System Concepts (10th Edition). Wiley Global Education US. https://online.vitalsource.com/books/9781119320913

What is the Difference Between Protection and Security - Pediaa.Com. (2018). Retrieved 1st August 2022, from https://pediaa.com/what-is-the-difference-between-protection-and-security/#:~:text=Protection%20and%20Security-,Definition,proper%20functioning%20of%20the%20system.

Police-reported cybercrime, number of incidents and rate per 100,000 population, Canada, provinces, territories, Census Metropolitan Areas and Canadian Forces Military Police. (2022). Retrieved 10 August 2022, from https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=3510000201&cubeTimeFrame.startYear=2017&cubeTimeFrame.endYear=2021&referencePeriods=20170101%2C20210101

Access matrix in Operating System - GeeksforGeeks. (2019). Retrieved 23 August 2022, from https://www.geeksforgeeks.org/access-matrix-in-operating-system/#:~:text=Access%20Matrix%20is%20a%20security,domains%20and%20columns%20represent%20objects.

Authentication and encryption in the 1Password security model
https://support.1password.com/authentication-Published: August 19,2022 from
encryption/#:~:text=Encryption%20transforms%20meaningful%20data%20into,that%20you%20know%20a%20secret.

CPU Scheduling and Virtualization

Abstract:

Within this paper CPU scheduling algorithms and Virtual machine algorithms are compared and contrasted with methods of solving them and effectiveness of said methods as well. The goal of my studies in this paper is to find the interconnecting topics of virtualization to the entire topic of Process management and related sub categories. This topic would be easier related to the storage topic and subtopics given the nature of virtualization, but I aim to learn something and it is my hypothesis that in digging I will find a surprising amount of inter-connectivity and at least one dead end within the subtopics. Virtual machines are taking a more heavy precedent over other virtualization concepts which may be able to cover these topics as well and possibly better, as virtual machines are of more specific interest and have a wider range of scholarly articles available.

Introduction/Background:

The motivation behind the research within this paper is to uncover interrelations between the 2 separate broad categories of process management and virtualization. While breaking down this topic into specifically OS process management, and comparatively the process management of Virtual machines.

Process management is an important aspect in operating systems. Processes need certain resources such as I/O devices, CPU time, memory and files to carry out the process, and is usually done during execution. A process is simply yet broadly a program loaded and executing into memory. Processes can also be referred to as jobs, yet this is an outdated term but it gives us a different perspective on what it is we are talking about. The value of the program counter provides the status of the processes activity along with the contents of the processors registers. A process can be represented by text section(executable code), data section (global variables), heap section (memory that is dynamically allocated during run time), stack section (temporary data storage when invoking functions).(Silberschatz, A, 2018)

CPU Scheduling is done by switching the CPU among processes, which can eliminate idle time, and increase efficiency overall. It can be broken into 2 parts, thread scheduling and process scheduling. On multi core computers many different process and threads can be running at the same time, the idea is to keep the system busy continuously through this system. (Silberschatz, A, 2018)

Synchronization is the coordination of access to data by multiple threads or processes. Control of access to shared data among processes through process synchronization is achieved by carefully using certain tools. The main goal of process synchronization is to prevent something called race condition. The race condition is a situation where two threads are concurrently attempting to change the value of a single variable. (Silberschatz, A, 2018)

Deadlock handling is done through a variety of methods. A deadlock is when a thread enters a waiting state due to needed resources not being available but the resources are being held indefinitely by other threads, creating a frozen state. Deadlocks can be combated using 2 different sets of techniques which are deadlock detection and deadlock prevention. (Silberschatz, A, 2018)

Virtualization was first provided by the IBM VM operating system. This system has since evolved and revolutionized virtualization, and is still available for use. In addition, many of its original concepts are found in other systems still.Virtualization is a broad topic in which virtual machines fit into. The idea of a virtual machine being to make a computer capable of doing specific things with a different kind of operating system that the computer in use virtually available and switchable between the host computer and the Virtual one. This virtual machine manager (VMM) is also known as a hypervisor, which creates and runs virtual machines by providing an interface identical to the hosts. (Silberschatz, A, 2018)

Methods

The problem being how CPU scheduling affects virtualization is a tricky topic to tackle as it is very specified and there are already limited sources within the framing of a 5 year time radius in scholarly peer approved journals. The problem can be tackled by refining searches by instead broadening them then looking within these texts for more evidence of such findings that support the ideas of how cpu scheduling and virtualization interact with each other. So with figuring out that there is VCPU scheduling and CPU scheduling I can refine search results and then become in line with exploring the topic I had otherwise never considered. There is talk of scheduling algorithms which I will tackle briefly but mainly diving into what VCPU scheduling is and the interactivity between it and the host CPU's scheduling algorithms.

The lockholder problem in VM vCPU scheduling is a result of when VM workloads have parallel applications like multithreaded programs, and programs with synchronous operation the proportion sharing algorithm results in poor performance. This is because of the task scheduling within the virtual environment being in 2 layers, VM internal process scheduling and VMM scheduling VCPU. The parallel workloads need a lock to synchronize. The VMM can preempt the running vCPU to run the other vCPU which may result in a vCPU holding a spin lock being descheduled, where a future vCPU needs this spinlock.(Yu,L, 2020) This is a problem that vCPUS reach simmilar to deadlock conditions of held processes or threads.

Parallel programming can be aided by use of virtualization. "In the past, without virtualization, in order to effectively solve the scheduling problem of parallel workload, Ousterhout18 proposed co-scheduling algorithm, which can simultaneously schedule concurrent threads to reduce synchronization delays."(Yu,L, 2020)

That being said, Virtualization also makes synchronization delay problems worse. Preemption of spin locks happens in the guest OS which cannot occur in non virtualized environment. "In order to reduce the synchronization delay of SMP VM, a lot of work27-30 propose co-scheduling where VCPU is treated as a process, and all

VCPUs of a VM are scheduled simultaneously. Co-scheduling under virtualization is not fresh. It can schedule multiple computational jobs (VCPUs, processes or threads) and is expected to reduce the execution time of parallel applications. "(Yu,L, 2020)

It is clear that virtual machines while providing some luxury to problem solving can also result in some complex computational issues as well.

Generally the high availability, low cost, on- demand service, and scalability of cloud computing makes it a popular choice for many OS related needs.

"Data center uses virtual machines to schedule tasks and uses the isolation of virtual machines to provide users with good security. At the same time, the redundancy technology of the data center ensures the security of the data stored in the cloud.

Even if a certain node or a certain area is paralyzed, the standby node in another area can replace the failed node at the first time."(Liang, B, 2022) Further reinstating the reliability of virtualization methods, which leads us to believe that while problems of complex nature can occur do they really happen that often and what can we do to prevent them? Without diving too in depth into the mathematical equations of algorithms used we can look at preventative methods of lock conditions in VMs and explore how effective they are.

"Multiprocessor virtual machines encounter synchronization problems such as lock holder preemption (LHP) and lock waiter preemption (LWP). When the issue happens, a virtual CPU (VCPU) waiting for such locks spins for an extraordinarily long time and wastes CPU cycles seriously, resulting in a significant degradation of system performance."(Yu, C, 2019) It is the multiprocessor virtual machines who cause these problems specifically. Co scheduling can alleviate said conditions, lock LHP avoidance, and also waiter yielding. Although these solutions are known to have short comings as well. While spinlock happens in native OS the avoidance techniques are seemingly more effective (Yu, C, 2019).

Results and findings:

CPU and vCPU Scheduling in Virtualization

In virtualization the multiplicity of CPUS/vCPUS (virtual CPUS) requires CPU scheduling for efficient load balancing of the system. Designing a CPU scheduling approach that can lead to efficient utilization of both vCPUs and CPUs can be achieved. In order to do this it is needed to balance the load of active domains in the vCPU and also CPU core level. A proposed model of scheduling is "Allocation of workload to each domain and initial mapping of vCPUS to pCPUcores, Calculation of load at vCPU level planning unpinning of vCPUs based on predefined threshold, Calculation of load at pCPU mapping. Relocation of vCPUs for load balancing"(MOHARANA, S. C, 2018).

CPU co-scheduling in a virtualization environment

"The VMM (virtual machine manager) scheduling algorithm asynchronously schedules each VCPU (virtual CPU) of a VM and ensures the CPU time usage of each VM. This proportional share method is widely used, because it simplifies the implementation of VMM CPU scheduling algorithm and can provide near-perfect performance for most ordinary workloads. However, when a VM runs with parallel workloads, the above method causes performance degradation because of the negative impact of virtualized systems."(Yu, L, 2020) Task scheduling within the virtual environment is spilt into two categories,the first being the VM internal process scheduling, while the other is VMM scheduling VCPU. Process scheduling is transparent to the VMM, and the VMM determines whether the current VCPU may be scheduled just by seeing any task need to be run on the VCPU.(Yu, L, 2020)

Findings

Where I have managed to find scheduling sources combined with Virtualization in search results of scholarly articles it was seemingly impossible to find the other topics which were listed above but I managed to find a few sources talking of "Virtualization Management" which is likely a completely different thing from process management at a cursory glance, I temporarily refine my focus to diving deeper into the realm of virtualization management. Furthermore upon reading through them realized this wasn't an overly interesting topic compared to the one I had stumbled upon seemingly by accident which is *How CPU scheduling affects Virtualization, or vice versa.*

I managed to come around full circle and connect concepts of virtual machines and their scheduling within the entire specified scope of process management. I thought perhaps initially that Virtual machines had their own definitions on how they perform and had some misconceived ideas about how they go about handling processes comparatively to operating systems, but this makes sense in the end clearing the fog that virtual machines function like operating systems do but virtually. Then after this realization I had more exploring to do of the differences between virtual machines handling of processes and the host operating system, and also maybe how they interactively cooperate or cause complications.

Further refining my findings I did not find what I set out to in the previous paragraph but taking a look back from where I started and the maze of sources it is good to have uncovered relational topics between 2 broad separate groupings of topics and explore some of the intricacies between them.

Laboratory Virtualization management

"The school uses Microsoft RDS virtual desktop technology to improve and manage the laboratory in a centralized manner, and the design is based on the total scale of 800 users and 500 concurrent users in the early stage [[12]]. The virtual desktop user resource configuration is as follows: 4G memory, 2vCPU, 50 GB system space, 10 GB user data storage space, and resource expansion will be carried out gradually. Under the condition that the management design structure remains unchanged, it will eventually meet the needs of 5,000 people. After the platform is completed, the unified management of the entire laboratory terminal will be realized, the local terminal will only be used for connection use, and the computing tasks will be transferred to the cloud desktop, which can effectively improve the service life of the local terminal, so as to realize the reduction of TCO (total cost of ownership)."(Liu Y, 2022). So the objective being to convert an entire laboratory database to cloud architecture, which is previously only locally hosted storage.

An efficient virtual CPU scheduling in cloud computing

"In cloud computing, fine-grained virtual CPU scheduling techniques are essential in hiding physical resources from running applications and mitigating the decrease in performance upon virtualization. However, evaluating and predicting the behaviors of virtual processors is getting harder because of the diverse QoS requirements of cloud applications."(jang, j, 2020) Examples of VMM's include Hyper-V, VirtualBox, Xen, and KVM.

Related work

Whereas the findings section partly details excerpts that explain what those sources I have included are generally trying to achieve the related work section details the works that are related to the topics discussed by those sources.

The related work to that which I have researched is mainly the attempt to make an algorithm which will solve both LHP and LWP and simmilar types of scenarios where it is explained that the conditions of complications which arise in cloud computing are ineffectively solved by current algorithms and this creates the need for another to better handle this process. (Yu,L, 2020),(Liang, B, 2022).

Other related articles would be anything mentioning virtualization as one can broadly extract information from these sources and also the textbook definitions of virtualization and process management and all its subtopics(Silberschatz, A., 2020)

Conclusions and future work

Future work on this topic could be boiled down into more specified effectiveness of virtualized scheduling algorithms, how those algorithms can be improved through process management. One clear conclusion I have made from my original hypothesis is I wasn't able to pinpoint the interrelation between OS scheduling and VM scheduling if there were any conflict between them at all, but it is also highly possible that they have little to no interaction as the idea of a virtual machine states it is entirely its own machine but virtually hosted by the native OS, so they may function entirely independantly where the scheduling of CPU processes don't ever exactly interact with vCPU.

Processes of both virtual machines are distinct and problems that occur may be shared amongst native OS and VM alike but the way the system handles them is different and the way in which they occur is also different.

Research paper references

Silberschatz, A., Gagne, G., & Galvin, P. B. (2018). Operating System Concepts (10th Edition). Wiley Global Education US. https://online.vitalsource.com/books/9781119320913

MOHARANA, S. C., SAMAL, S., SWAIN, A. R., & MUND, G. B. (2018). Dynamic CPU scheduling for load balancing in virtualized environments. *Turkish Journal of Electrical Engineering & Computer Sciences*, *26*(5), 2512–2524. https://doi.org/10.3906/elk-1709-112

Yu, L. (2020). CCHybrid: CPU co-scheduling in virtualization environment. Concurrency and Computation,32(3).https://search.ebscohost.com/login.aspx?direct=true&db=edsbl&AN=vdc.100097876059.0x000001&site=eds-live

Liu, Y., Duan, S., & Wang, B. (2022). Laboratory Virtualization Management Based on Deep Integration of Cloud Desktop. Wireless Communications & Mobile Computing, 1–7. https://doi.org/10.1155/2022/4487023

Jang, J., Jung, J., & Hong, J. (2020). An efficient virtual CPU scheduling in cloud computing. Soft Computing - A Fusion of Foundations, Methodologies & Applications, 24(8), 5987–5997. https://doi.org/10.1007/s00500-019-04551-w

Liang, B., Liu, R., & Dai, D. (2022). Design of Virtual Machine Scheduling Algorithm in Cloud Computing Environment. Journal of Sensors, 1–8. https://doi.org/10.1155/2022/3848165

Yu, C., Qin, L., & Zhou, J. (2019). A lock-aware virtual machine scheduling scheme for synchronization performance. Journal of Supercomputing, 75(1), 20–32. https://doi.org/10.1007/s11227-015-1557-y