Comp 314: Graphic Design
ID: 3441988

Andrew Mackay

Operating systems assignment 2

1. a The short term scheduler refers to the “Ready queue” whereas the long term scheduler is the “Waiting queue” as the process is Ready to be further executed within the ready queue whereas the other is still waiting.

b. The long term scheduler runs much less frequently than the short term scheduler, The short term scheduler typically runs every 100 milliseconds. So with that being said, at least twice that time is when the long term scheduler should run,

c. The interrupt can occur between the ready, waiting, and running elements of the diagram which could include any type of operation deemed as more urgent like input from a user or a system process deemed more important to execution within that time frame.

d. Within the process diagram a I/O request will cause an interrupt where the process will move from running to waiting, for the Input output completion. When the request is completed the process is moved to the ready state and resumes its functioning from that standpoint.

e. The exit() system call is used by the OS for process termination where a process can be terminated after being executed.

f. Processes in the android system have an order of importance where if the user is using a process in the foreground it is of the utmost importance as that is what is directly related to what the user is currently doing. Then a visible process which is a process that is visually effecting something the user is doing like running an activity or service. Then a service process which is holding a service, which if services have been running for too long will be demoted in importance. Whereas the final process is a cached process so that is the bottom of the command line and can be terminated if needed.(developer.android,2022) This makes a fair amount of sense based on the hierarchy of importance and would be hard to argue otherwise based on user experience.

2.

A context switch is performed by a state save, performed by the kernal, of the current process and a state restore of a separate process, When responding to an interrupt, the OS saves the current context running on the CPU, then the attention is brought to what caused the interrupt while the other original process is suspended which is later returned to after the interrupted process is executed.

The process control block, also known as “PCB” is a data structure that contains information of related processes. This is an important representation of data which includes process state, process number, program counter, registers, memory limits, list of open files, CPU scheduling info, memory management info, I/O status info, Accounting info, and Location of the process control block.(tutorialspoint, 2022). During a context switch the kernal saves the old process in its PCB then loads the saved context of the new now current process.Context switch time varies from machine to machine and is highly dependant on hardware support, if a machine has more registers this could be a advantage to saving time during a context switch although it depends on the special instructions of the processes carried out, but typically this is still achieved within milliseconds.

3.

There is two fundamental models of IPC: Message Passing and Shared memory. Both are somewhat opposite yet share the same end goal.

Shared modal is a region of memory that is used for communication between processes where they share a common address. Written explicitly by the application programmer, since done through shared memory speed is greatly optimized, as system calls in this case are only made to establish the initial shared memory.

In Message passing modal communication takes place in a distributed environment by means of message exchanged between the co-operating processes and it is slower as compared to shared memory method. This is because it makes use of system calls to provide communication between processes, otherwise known as kernal intervention. (geeksforgeeks,2020).

4.

FCFS stands for first come first serve and it represents the type of process scheduling where a set of processes will stack each other in order of when they requested the CPU. So if a very long process enters the CPU it cannot be removed until it is complete. Which then leaves short I/O processes to wait a long time, and this can impact performance of the PC. The more long processes that are at the head of the queue the longer the short and easily executable processes will have to wait.

Even thought first come first serve is the most straight forward process scheduling technique it becomes very inefficient for this reason as short processes that could be executed in fractions of a second then can take multiple seconds before being executed.

5.

In scheduling a process from the ready state is loaded into running state. The selection of the process is done using various scheduling algorithms.

The goal of a scheduling algorithm is to maximize performance. The performance can be measured in two ways: Maximum CPU utilization: To maximize the CPU utilization it is insured by the operating system to keep the CPU busy for most of the time. So, when more processes are scheduled the CPU utilization gets increased.

Maximum Throughput: Throughput is the number of processes executed per unit time. So, to maximize throughput more processes should be scheduled to CPU.When throughput increases, CPU utilization also increases.Minimum Waiting Time: The amount of time the process is in the waiting queue should be less. Otherwise, it can lead to starvation.Minimum Turnaround Time: Turnaround time is the period between the completion time and arrival time.

6.

A process is both an active program and a program that is being executed. It is more than the program code as it includes the program counter, process stack, registers, program code etc. Compared to this, the program code is only the text section. Some benefits of processes over threads are that if a process gets blocked remaining processes can still execute as they are totally independent of one another,

A thread is a lightweight process that can be managed independently by a scheduler. It improves the application performance using parallelism. A thread shares information like data segment, code segment, files etc. with its peer threads while it contains its own registers, stack, counter etc. Benefits of a thread include them being lighter than processes while taking less time for context switches, Communication between threads are quicker than that of processes, threads share memory among one another, and the time for creation and termination of threads is less.(2022, tutorialpoint).

7.

When several threads are competing for few resources, a thread will then enter a waiting state, but sometimes, a waiting thread can never again change state, because the resources it has requested are held by other waiting threads. This situation is called a deadlock. Deadlock occurs when every thread in a set is blocked waiting for an event that can be caused only by another thread in the set. There are several conditions that create a deadlock such as Mutual exclusion, Hold and wait, No

preemption, and Circular wait. 2 different kinds of deadlock prevention that may alleviate these conditions are to eliminate Hold and Wait, or to focus on eliminating the No Preemption deadlock.

Prevention from a process holding one or more resources while waiting for others is necessary to eliminate this kind of deadlock. It can be done by requiring all processes holding resources must release them before requesting new ones, then reacquire the resources with the new ones in one new request. Also by making a requirement that all processes request all resources at one time can eliminate the problem as well.(uic.edu, 2022)

For the no preemption deadlock it can be prevented by when a resource is requested and not available, then the system searches for what other processes currently have those resources and are themselves blocked waiting for another different resource. If said process is found then some their resources might get preempted and added to the list of resources that the process is waiting on(uic.edu, 2022).

8.

Deadlock detection differs from deadlock prevention in the literal description of the terms that one is trying to find and locate the deadlock whereas another is attempting to prevent a deadlock from ever occurring. Although deadlock detection doesn't simply just detect a deadlock it is an approach on its own to eliminating a deadlock as well. Real time OS implement deadlock recovery. Manual user intervention can aid recovery from a deadlock as well as Process termination, and Resource preemption

Terminating the process or processes until the system recovers from the deadlock is one method of recovery but may result in unnecessary deletion of processes.

Preemption of resources is the second option for deadlock detection/recovery and it is carried out by deciding which resource to preempt from which processes, then roll back the preempted process to a safe state although the only safe roll back is at the beginning of the process. Lastly, starvation is prevented by giving priority to the processes that get preempted and every time they get preempted again it increases the priority that much more so eventually it will be pushed to the front of the queue.((uic.edu, 2022).

9.

In the Readers-Writers problem database which is shared among several concurrent processes, some of these processes want to read whereas others want to both read and write, which is otherwise known as update. If a writer process and reader try to access the database at the same time this is where the problem arises. Starvation is a situation which occurs when a process or thread is stuck waiting indefinitely in queue to be dealt with. This can occur within the readers writers problem in 2 separate ways. The first is that there is a requirement that no reader be kept waiting for other readers to finish because of a waiting writer. The second is that

if a writer is waiting for access to a object , there can be no new readers that can start reading.

10.

While a multiprocessor is a CPU chip with multiple processors within it. An asymmetric multiprocessor is a scheduling algorithm in which a boss processor controls the system and the other processors have their own tasks or receive further instruction from the boss processor, and only one processor accesses the system data structures while others run user threads, reducing the overall need for any data sharing.

With a symmetric multiprocessor all tasks within the multiple processors are distributed equally. Which is also runs on its own a multiprocessor scheduling method, where each processor can run kernel threads or user-level threads, and is ultimately self-scheduling.

11.

A thread can also be called a lightweight process. They are broken into two different types of categories, user level-thread and kernel level thread whereas user level threads are managed by users and kernal threads are managed by the OS. Multi threads combine these 2 types of threads into a category called multi level threading.(tutorialspoint, 2022). Multicores are essentially built to handle multithreaded processes but there are extensive design considerations for engineers when implementing this system and is quite complex yet still worth it in the efficiency it creates. The types of programming challenges for multicore processors are identifying tasks that are divided into concurrent separate tasks , balancing the separate parallel tasks to perform equally, Splitting the data to run on separate cores, the data dependencies must be taken into account so that tasks can be run in a synchronized way, and the testing and debugging on multiple cores and multiple paths is more complex than single threaded debugging.

12.

Fine-grained multithreading is a mechanism where a chache miss due to thread instruction will not affect the switch of threads.(pedia,2019). The fine grained threading switches between threads at a much higher level of granularity, or in other words can be thought of to be the more thorough of the 2 types. Course-grained multithreading executes a thread on a core until a long latency event occurs which forces a switch between threads but this is a more lengthy process as the pipeline needs to be flushed before the other thread can begin being executed.

13.

Within deadlock detection techniques the wait for graph is used to indicate a deadlock in operating systems if there is a cycle happening within the graph. This graph is a variation of the resource - allocation graph, which is made by removing the

resource nodes of the resource allocation graph which creates an arrow shaped module of T1-T5. The system needs to periodically invoke an algorithm to detect if there is a cycle using the graph. If the wait for graph contains a cycle then it is known there is a deadlock occurring.

The hold and wait condition is referring to what is happening during this kind of deadlock, the process is holding onto resources that may be trying to be used by another process. If deadlocks are occurring in the system frequently, than this process should be invoked frequently as well. One disadvantage of invoking the detection is the excessive wait time caused by the request for allocation in memory(high-tech-guide,2022)

References

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

Processes and Application Lifecycle | Android Developers
Processes and Application Lifecycle | Android Developers. (2022). Retrieved 14 August 2022, from https://developer.android.com/guide/components/activities/process-lifecycle

Difference between Process and Thread
Difference between Process and Thread. (2022). Retrieved 8 July 2022, from https://www.tutorialspoint.com/difference-between-process-and-thread#:~:text=an%20active%20program.-,A

Operating Systems: Deadlocks
Operating Systems: Deadlocks. (2022). Retrieved 9 July 2022, from https://www.cs.uic.edu/~jbell/CourseNotes/O

Difference between Shared Memory Model and Message Passing Model in IPC - GeeksforGeeks

Difference between Shared Memory Model and Message Passing Model in IPC - GeeksforGeeks. (2020). Retrieved 16 August 2022, from https://www.geeksforgeeks.org/difference-between-shared-memory-model-and-message-passing-model-in-ipc/

What is Process Control Block (PCB)?
What is Process Control Block (PCB)?. (2022). Retrieved 16 August 2022, from https://www.tutorialspoint.com/what-is-process-control-block-pcb

Operating System - Multi-Threading
Operating System - Multi-Threading. (2022). Retrieved 16 August 2022, from https://www.tutorialspoint.com/operating_system

What is the Difference Between Fine Grained and Coarse Grained Multithreading - Pediaa.Com. (2019). Retrieved 5th August 2022, from

https://pediaa.com/what-is-the-difference-between-fine-grained-and-coarse-grained-multithreading/#:~:text=The%20main%20difference%20between%20fine,instructions%20until%20a%20stall%20occurs.

WHAT IS MEANT BY WAIT-FOR GRAPH? high-tech-guide.com. Available at: https://high-tech-guide.com/article/what-is-meant-by-wait-for-graph [Accessed August 3, 2022].