**Assignment 2**

**1、Assume a computer contains 1024MB of main memory, if an operating system requires a virtual memory of twice that size using pages of 4KB, how many pages would be required?**

**Answer：**

virtual memory size = 2 \* main memory size = 2 \* 1024MB = 2 \* 1024MB \* 1024 = 2097152KB

number of pages = virtual memory size / page size = 2097152KB / 4KB = 524288

So, 524,288 pages are needed to accommodate a virtual memory of twice the size of the 1024MB main memory using 4KB pages.

**2、If a multiprogramming operating system allocates time slices of 20 milliseconds(毫秒) and the machine executes an average of 5 instructions per microsecond(微秒:百万分之 一秒). How many instructions could be executed in a single time slice?**

**Answer：**

20 ms = 20,000 µs

instructions = (instructions per µs) x (time in µs) = 5 instructions/µs x 20,000 µs = 100,000 instructions

So, in a single time slice of 20 milliseconds (20,000 microseconds) with an average execution rate of 5 instructions per microsecond, the machine can execute 100,000 instructions.

**3、A process is said to be I/O-bound if it requires a lot of I/O operations, whereas a process that consists of mostly computations within the CPU/memory system is said to be computebound. If both a compute-bound process and an I/O-bound process are waiting for a time slice, which should be given priority? Why?**

**Answer：**

In this scenario, priority should be given to the I/O-bound process. This is because I/O-bound processes typically require a significant number of I/O operations, which may involve waiting for responses from external devices. If these I/O operations are not performed promptly, processes may end up waiting for extended periods, wasting CPU time.

In contrast, compute-bound processes primarily perform computational tasks and generally do not require waiting for external device responses. These processes typically use the CPU for extensive computational operations, and their waiting times are relatively minimal.

**4、Since each area on a computer’s display can be used by only one process at a time (otherwise the image on the screen would be unreadable), these areas are nonshareable resources that are allocated by the window manager. Which of the three conditions necessary for dead lock does the windows manager remove in order to avoid deadlock?**

**Answer：**

The windows manager can remove the first condition: competition for non-sharable resources in order to avoid deadlock.

**5、If a typist types 60 words per minute (where a word is considered five characters), how much time would pass between typing each character? If a multiprogramming operating system allocated time slices in 20 millisecond units and we ignore the time required for process switches, how many time-slices could be allocated between characters being typed?**

**Answer：**

First, the typist types 60 words per minute, so the time taken to type one word is:time per word = 1 minute / 60 words = 60 seconds / 60 words = 1 second / word

Then, a word is considered five characters, so the time taken to type one character is time per character = time per word / 5 characters = 1 second / 5 characters = 0.2 seconds / character

Next:20 milliseconds = 0.02 seconds

Last: number of time slices between characters = time per character / time per time slice = 0.2 seconds / 0.02 seconds = 10

So, there will be 10 time slices allocated between characters being typed.

**6、What problem arises as the lengths of the time slices in a multiprogramming system are made shorter and shorter? What about as they become longer and longer?**

**Answer：**

Short Time Slices:

* Overhead: If time slices are made very short, the overhead associated with context switching between processes becomes more significant. When time slices are very short, a significant portion of CPU time may be spent on context switching, which can reduce the overall system efficiency.
* Low Throughput: Short time slices can result in lower throughput because processes are frequently interrupted to give a chance to other processes. This can lead to reduced overall system performance.

Long Time Slices:

* Resource Hogging: Long time slices can lead to resource hogging by processes. If a process is given a long time slice and it doesn't voluntarily yield the CPU, it can monopolize system resources, making it challenging for other processes to get CPU time.
* Increased Latency: Long time slices can introduce latency in responding to external events or interrupts because the system may not check for new events until the current time slice is completed.