

Unit OS4: Scheduling and Dispatch

4.7. Quiz

Windows Operating System Internals - by David A. Solomon and Mark E. Russinovich with Andreas Polze

Copyright Notice

© 2000-2005 David A. Solomon and Mark Russinovich

- These materials are part of the *Windows Operating System Internals Curriculum Development Kit*, developed by David A. Solomon and Mark E. Russinovich with Andreas Polze
- Microsoft has licensed these materials from David Solomon Expert Seminars, Inc. for distribution to academic organizations solely for use in academic environments (and not for commercial use)

Windows Operating System Internals - by David A. Solomon and Mark E. Russinovich with Andreas Polze

Optimization Criteria

The scheduler in an operating system has to manage the CPU efficiently. Which one of the optimization criteria mentioned below is not related to CPU scheduling?

- a) Maximum CPU utilization
- b) Maximum throughput
- c) Minimal turnaround time
- d) Minimal waiting time
- e) Minimal response time
- f) Maximum Mean Time To Failure (MTTF)

Processes and Threads

Which information is not associated with a thread's Thread Control Block (TCB):

- a) Program counter
- b) CPU registers
- c) Memory management information
- d) CPU scheduling information
- e) Pending I/O information

Multiprocessing

The Windows OS family supports multiple CPUs. Which approach is taken by the Windows scheduler?

- a) Symmetric multiprocessing
- b) Asymmetric multiprocessing
- c) Cooperative multiprocessing

Process Creation - Windows API

In Unix, new processes are created using the `fork()` system call. New programs can be loaded into an existing process using `exec()`. How does the Unix approach differ from the `CreateProcess()` system call found in the Windows API?

- a) `fork()/exec()` allows for input/output redirection, `CreateProcess()` does not
- b) `fork()/exec()` allows for setting up pipes, `CreateProcess()` does not
- c) With `exec()` the process identifier remains unchanged, with `CreateProcess()` it changes always
- d) `fork()` implements copy-on-write, `CreateProcess()` does not

Windows OS Scheduling

Which of the below-mentioned attributes does not fit to the Windows OS scheduling algorithm?

- a) Cooperative
- b) Preemptive
- c) Round-Robin

Scheduling Algorithms

Which one of the below-mentioned operating systems uses a decreasing-priority-Scheduler for a fair-share management of the CPU?

- a) MS-DOS
- b) Windows NT/2000/XP
- c) Berkeley UNIX 4.2 BSD
- d) VMS

Windows Scheduler

The Windows OS kernel scheduler manages:

- a) Threads
- b) Fibers
- c) Processes

Threads and Processes

Windows NT/2000/XP -
Which statement is true?

- a) Threads have exactly one Stack
- b) Each process contains at least one thread
- c) All threads share security settings of the process they are belonging to

Windows Processes

For a Windows process, the OS maintains various data. Which information is associated with the executive process object?

- a) Access Token
- b) List of Virtual Address Descriptors (VADs)
- c) List of Threads
- d) List of open handles
- e) List of open windows

Processes & Threads

Which of the arguments listed below is not related to dividing an application into multiple threads?

- a) Better user responsiveness, decoupling of GUI and background execution
- b) Take advantage of multiple processors
- c) Increased application complexity
- d) Better usage of installed main memory

Quantum

The prolongation of the quantum value used in the implementation of a scheduling algorithm has the following goal:

- a) Increase of throughput
- b) Optimization of interactive response time
- c) Increase of overall system stability
- d) Guarantee of real-time behavior

User-Space vs. Kernel-Threads

The Windows OS family implements multi-threading. Which one of the below-mentioned approaches is not present in the Windows OS?

- a) One-to-one mapping of user-space-threads onto kernel-threads
- b) One-to-many mapping of user-space-threads onto kernel-threads
- c) Many-to-one mapping of user-space-threads onto kernel-threads

Priority Elevation

The Windows OS scheduler occasionally increases the priority of non-real-time threads. What is the priority of a waiting GUI thread after the occurrence of a GUI event (window message)?

- a) 15
- b) Depending on the actual device driver involved
- c) 31
- d) Current priority is increased by 2 (but never exceeds 15)

Windows priority levels

Windows has a total of 32 thread priorities. Which one of the statements below is false:

- a) The highest priority is 31
- b) Priorities above 15 are referred to as "realtime" priorities
- c) Priorities below 16 are adjusted dynamically by the scheduler
- d) Increasing a thread's priority will make the thread run faster - even on an otherwise empty system

Running a program in Windows

Which type of programs can be run natively in 32-bit Windows without requiring a helper image?

- a) MS-DOS programs
- b) Windows 32-bit programs
- c) Posix programs
- d) OS/2 programs
- e) Windows 16-bit programs