Fyrirlestraræfing 1

1. What is an OS interrupt and how is it defined?

A. Operating systems are interrupt driven. The interrupt transfers control from any instruction that is currently executed by the CPU to the interrupt service routine. <u>RÉTT!</u>

B. OS interrupt is always initiated by the OS system t.b handled within the OS Kernel

C. OS interrupts are only use within asynchronous computer systems and are initiated by the user

D. Non of the above answers are correct

2. What is the purpose of having timer for interrupt control?

A. Timer are not used by OS interrupts

B. Timer is used to switch between processes that are executing by the CPU <u>RÉTT!</u>

C. OS interrupt timer is always set to 20ms which is the time it takes to execute CPU micro program routines

D. both b) and c)

3. What a Mode-bit within OS and give an example of usage?

A. Mode bit is firmware (HW) related and not used by the OS system

B. Mode bit are used for swapping Main memory to external storage during Memory management

C. Hardware dedicated bit. Showing if the CPU is doing a task on behalf of the user or the OS kernel <u>RÉTT!</u>

D. The mode bit is set within the OS kernel to indicate an active OS interrupt

4. Explain roughly the purpose of Cache within OS?

A. Used within servers to store redundancy data

B. Cache is used within CPU to store copied information t.b accessed "fast" information

C. To store frequently re-referenced information that can be accessible “fast”.

D. both b) and c) <u>RÉTT!</u>

Fyrirlestraræfing 2

1. How are OS services defined

A. Services that are helpful to users, applications and system operations. <u>RÉTT</u>
B. Services provided for the OS by user applications
C. Middlware system services provided for the OS
D. System services used by OS and user applicatison

2. What are system calls used for?

A. Interaction between OS upper layer and the OS kernel
B. A way for programs to interact with the OS
C. They are OS calls that include software interrupt Traps
D. both b) and c) <u>RÉTT!</u>

3. Name three main types of OS architecture mostly used today?

A. Monolithic-, Layered- and Microkernel OS
B. Monolithic modules- Microkernel- and Layered OS
C. Module based OS , Layered- and Microkernel OS. <u>RÉTT!</u>
D. Module based OS , Monolithic- and Microkernel OS.

4. Why is it better to use Virtual Machines (VM) in given circumstances

A. Using VM in multitasking environment is more reliable and robust
B. It reduces overload as VM processes can communicate directly between VM machines
C. A VM allow us to run OS's on a given machine while making it still think it’s OS is isolated on that machine <u>RÉTT!</u>
D. Non of above

5. What is the purpose of Hypervisor within VM?

A. The Hypervisor is used to protect the host VM OS from host operative systems failures
B. It allows guest OS to support multiple guest VMs by virtually sharing its resources.
C. Hypervisor creates and executes VM and schedules CPU time between OS's. We should use VM to protect the hardware and software of the host system. It allows host computer to support multiple guest VMs by virtually sharing its resources. <u>RÉTT!</u>
D. both a) and c)

Fyrirlestraræfing 3

1. What is the purpose- and context of the PCB?
A. The PCB is used by the OS to locate running processes. It stores informations about valid processes
B. It stores the process state id´s, program counter, I/O status, Accounting- and CPU scheduling info. It is used for saving and rescheduling a running process. <u>RÉTT</u>
C. It stores information about CPU-schedular algorithms and used by CPU-schedular to select scheduling strategy
D. The OS kernel uses the PCB for memory management where it includeds information about available sweet of memory addresses within Main memory.

2. What type of interrupt would relate to process “wait” state?
A. Only HW interrupt
B. I/O interrupt <u>RÉTT!</u>
C. All Trap interrupts
D. Kernel Mode bit

3. What is in control of a child process if parent process terminates unexpectedly within POSIX (e.g. Unix) OS?
A. The CPU registers
B. Nothing as the child process would die along with the parent process
C. The child process becomes a new parent process having it´s own control
D. The init process (Pid=1) <u>RÉTT!</u>

4. How are the Ready queues and Waiting queues implemented (roughly)?
A. With pointers and are mostly singly linked lists where queue header points to the first and last PCB's in to the list <u>RÉTT!</u>
B. The queues are implemented with recursive calls when calling the next PCB in line
C. Ready queue is implemented with recursive mechanism while the waiting queue uses linked lists where the queue header points to the first and last PCB's in to the list
D. Non of the above

5. When is the short term Low level/CPU-scheduler activated?
A. It is activated after/during SW Trap- and HW interrupt
B. Only Long term and Medium term scheduler can be activated during OS interrupt
C. When process gets blocked, time slices expires, I/O completes, process terminates
D. both a) and c) <u>Rétt!</u>

Fyrirlestraræfing 4

1. Name two types of IPC and their limitations?

A. Shared memory requires that the communication processes reside on the same machine. 2. Message passing where the processes need to establish a communication link between them and exchange messages via the com link using send/receive mechanism. Can only exchange small chunk of data <u>RÉTT!</u>

B. Combined shared memory and message passing. Have heavy performance issues between machines. 2. Shared processor memory between multiple machines but need to run the same type of OS

C. Using IPC communication threads but can be unreliable due to lack of synchronization. 2. Message passing without special establishing the com link. Mostly used with large data but expensive to used with small chunk of data

D. Non of above

2. What are Threads? Describe roughly items that are contained within Thread Control Block?

A. Threads are heavy weight processes where TCB contains everything within PCB plus thread specific memory information

B. The TCB is used for context switching by the schedular and they contain the PID of the process that is in the state "running"

C. Threads are light weight processes. TCB contains registers, stack counter, states and Program counter <u>RÉTT!</u>

D. Threads are equal to small processes and they contain only the current state of the process

3. What is the Consumer in assignment 8 (client or server)?

A. Consumer is always the server

B. There is no relation between client/server and consumer/producer

C. Client (although in 1-1 client/server communication it is hard to tell) <u>RÉTT!</u>

D. Non of above

4. What is a rough description on how to communicate between processes running on different machines having different operating systems?

A. Using OS specific (Non standard) which is tailored towards the type of IPC used for different operating systems

B. Using shared memory communication mechanism where the OS is not needed to interact

C. IPC uses standardized communication mechanisms e.g. message passing and the Internet Protocol (IP) and is heavily used in client-server systems. <u>RÉTT!</u>

D. Non of above

5. What are sockets used for?
A. They are type of data structure which is used by pipes IPC mechanism
B. Sockets are endpoints of communication where communication takes place between a pair of sockets. They are only used for Connection oriented communications
C. They are special programming model used for context switching e.g. the CPU schedular
D. They are generic programming model for non-local and local interprocess communication. Sockets are endpoints of communication where communication takes place between a pair of sockets. <u>RÉTT!</u>

Fyrirlestraræfing 5

1. What CPU scheduling algorithm is often used more than others in long-term scheduling (batch jobs)?
A. FCFS
B. SJF <u>RÉTT!</u>
C. RR
D. SRTF

2. Name multithreading (user threads vs. kernel threads) model used in most used OS today?
A. One-to-one <u>RÉTT!</u>
B. One-to-many
C. Many-to-one
D. Many-to-many

3. What is a dispatcher used for?
A. Part of the OS which is used for prioritization
B. Mechanism within the OS serving I/O controllers
C. Part of the CPU core which is used for load balancing of processes/ threads between available cores
D. The part of OS that performs the actual context switch b <u>Rétt!</u>

4. Which CPU scheduling algorithms Pre-emptive processes/threads and puts them at the end of the ready queue?
A. FCFS
B. SRTF and RR, with and without priorities. <u>Rétt!</u>
C. SJF
D. STS

5. What is one way of organizing scheduler queues in Symmetric Multiprocessing?
A. Processor/cores uses combined job queues and no ready queue needed but synchronization is needed between the cores
B. Every processor/core has its own ready queue which are “private” to each CPU/core and no synchronization needed for accessing the queues. <u>Rétt!</u>
C. Common public scheduling queue used by all cores an (Master/Slave cores) and is controlled by the master core before executing on any slave cores
D. Non of above

Fyrirlestraræfing 6

1. What can happen when processes/threads have shared memory?
A. They may loose part of their program data to a kernel mode process
B. The shared file system can be blocked
C. Race condition. Both threads access the memory at the same time and the expected result can be wrong RÉTT
D. The Main RAM memory may get blocked by TRAP system interrupt

2. What is the idea behind the critical section problem algorithm?
A. To make sure that many processes/threads are using (reading and writing) the same memory space at the same time.
B. To make sure that no two processes/threads are using (reading and writing) the same memory space at the same time. RÉTT!
C. To relocate part of the Main RAM memory to a secondary storage
D. It is used to control how the scheduler works

3. Why does the Peterson algorithm solve the critical section problem?
A. It blocks all other processes from accessing the CPU
B. It allows only two thread at a time to work within the critical section and activates a race condition.
C. It allows only one thread at a time to work within the critical section and prevents a race condition. RÉTT
D. Non of above

4. What are three main requirements for solving the race condition problem?
1. bounded waiting, mutual exclusion, progress RÉTT!
2. bounded waiting, starvation, progress
3. starvation, mutual exclusion, bounded running
4. bounded running, starvation, mutual exclusion

5. Can the race condition as defined in the book ch.6., occur without having shared memory area?
A. No RÉTT!
B. Yes
C. It is complicated and depends on the boundaries between the OS kernel thread and the process user thread
D. Both b) and c)

Fyrirlestraræfing 7

1. What are- and why can atomic instructions be useful in solving the critical section problem?

A. They use DMA for accessing the Main Memory and will not take extra CPU time
B. They automatically disable/enable interrupt with an special interrupt vector
C. They are not interrupted so they are optimal for processes working within critical section <u>RÉTT!</u>
D. Because they are loaded from application system library as default and do not require extra compiler time

2. Why is it not recommended to disable/enable interrupts when dealing with critical section problem in most OS today?

A. Because most OS's run on single-core/single processor
B. Because most OS'es run on multi-core/multiprocessor systems. This means that multiple processes can execute at the same time therefore running processes might enter a critical section even if interrupts are disabled <u>Rétt!</u>
C. The CPU scheduler can only handle interrupts within the same processor-core
D. None of above

3. How does the Semaphore algorithm solve the Busy waiting problem for the critical section?

A. It emits a signal that let sleeping/waiting process-threads continue executing when the critical section is free for access <u>Rétt</u>
B. It initiates a waiting state that let sleeping/waiting process-threads continue executing when the critical section is free for access
C. It request the CPU schedular to archive the process-thread at the end of the Ready que
D. Non of above

4. What does the value of semaphore from the semaphore "init" operation indicate?

A. It indicates how much "space" (in terms of number of processes that are allowed to enter) is left in the critical section <u>RÉTT!</u>
B. It turns off busy waiting for the critical section
C. It indicates the number of process/threads that are mutually excluded from the critical section
D. Both a) and b)

5. What are the main disadvantage when using programmed API semaphores?

A. It is easy to create wrong solutions using them which may lead to a deadlock of all involved processes
B. When performing a semaphore operations, system calls are required which are slow, because they involve, e.g., context switch which is time consuming
C. Multiple system calls are needed which can be expensive in terms of buzzy waiting
D. Both a) and b) Rétt!

Fyrirlestraræfing 8

1. When controlling access to a critical section then X can allow mutually exclusive access to a shared object but Y can allow multiple threads to access the shared object. What is- X and Y?

E. Y is a monitor, X is a semaphore
F. X is a monitor, Y is a semaphore <u>Rétt!</u>
G. X and Y are both monitors
H. Non of above

2. What is a rough description for the Monitor concept?

A. Monitor concept is the same as Semaphore but has it´s predefined operations declared within the programming language which implements the concept
B. Monitors control the dispatcher context switch used by the CPU schedular mechanism and set the flag for user/kernel mode
C. Monitor is a collection of variables and operations to access these variables. It activates wait to the calling process to sleep and signal to wakeup the process <u>RÉTT!</u>
D. Both a) and b)

3. What happens to a Monitor signal if there is no thread waiting?

A. An interrupt is emitted by the OS which the programmer need to control within a Try&Catch loop
B. The monitor counter will be increased which later could lead to a deadlock
C. The signal gets lost if there is no thread waiting <u>Rétt!</u>
D. The programming compiler is instructed to generate an error if there is no matching wait operation

4. What does the Java prefix “synchronized” represent?

A. A Java object has internally a lock flag and method that is declared as synchronized can only be executed if that lock is not set. <u>RÉTT</u>
B. It s a method declaration prefix which represent the return value from the method
C. It forces the compiler to to generate binary code for all operation within a method declared as synchronized
D. Non of above

5. Describe the Java Monitor operations used for condition handling within a critical section?

A. They are predefined methods (acquire(), release() and releaseAll()) within the top level base Thread class and can only be called within methods declared as not synchronized
B. They are predefined methods (wait(), signal() and signalAll()) within the top level base Thread class and can only be called within methods declared as synchronized
C. They are the predefined methods; wait(), notify() and notifyAll()) within the top level base Object class and can only be called within methods declared as synchronized . Rétt!
D. Non of above

Fyrirlestraræfing 9

1. How can we prevent that deadlock may occur?
A. Using OS specific tailored solution towards the type of deadlock
B. Always using Monitors to protect the sharable resources
C. By ignoring all possible deadlocks
D. By ensuring that one of four condition necessary for deadlock does not hold. 1. Mutual exclusion 2. Hold and wait 3. No preemtion 4. Circular wait. <u>RÉTT!</u>

2. When in your opinion should OS deadlock algorithm e.g "Matrix-based Deadlock Detection Algorithm" be triggered?
A. When the CPU utilization is high
B. When the CPU utilization is low
C. Randomly when more than two process are running
D. Non of above <u>EKKI RÉTT</u>

3. What method would you recommend to use for recovering the computer system from a deadlock situation?
A. Restart the operating system
B. Use matrix deadlock detection algorithm, e.g. the safe state or Bankers algorithm. Also possible to terminate deadlock processes according to given algorithm e.g. process priority <u>RÉTT</u>
C. Block deadlock processes for a period of time and then revoke them after other process have terminated
D. Non of above

4. The Safe state- and Bank algorithm are interconnected but how?
A. They come as predefined pair of the OS system command register
B. They both come from the matrix based detection algorithm where banker algorithm uses safe state strategy <u>RÉTT</u>
C. They are connected through the dispatcher Context switch
D. Both a) and c)

5. What are the main necessary conditions for a Deadlock?
A. Mutual exclusion, Hold and wait, Circular wait, No preemption <u>RÉTT!</u>
B. Non mutual exclusion, Exit and wait, Circular wait, No preemption,
C. Mutual exclusion, Hold and wait, Non circular wait, No preemption
D. Mutual exclusion, Hold and wait, Circular wait, Preemption

Fyrirlestraræfing 10

1. What are the benefits of having Dynamic-link Libraries?
A. It reduces the duplication of code and saves memory. As a result, the software and the OS runs more smoothly <u>Rétt!</u>
B. It protects the code from having linked errors although it increases memory. As a result, the software and the OS runs more smoothly
C. It prevents the OS from having external fragmentation within it´s processes
D. The only benefit is there is less code needed to be linked together when making an executable file

2. What are non-contiguous memory holes within a process/thread called?
A. External fragmentation
B. Separated holes
C. Internal fragmentation <u>Rétt!</u>
D. Non of above

3. Which MMU method will prevent having memory holes?
A. Swaped MMU´s
B. DMMU
C. Paged MMU´s <u>Rétt!</u>
D. Fragment MMU´s

4. How does the frame number algorithm transfer logical address space to physical address (you may give an example)?
A. logical address l=n*x/a (n=frame no, x=page size, a=offset)
B. physical address p=l*x+a (l=logical address, x=page size, a=offset)
C. physical address p=n*x+a (n=frame no, x=page size, a=offset) <u>Rétt!</u>
D. No of above

5. What distinguish External fragmentation and Internal fragmentation of memory holes?
A. The MMU uses medium term scheduler for External fragmentation and short term CPU scheduler for internal fragmentation
B. There is no distinction between the functionality of the two. Internal refers to OS processes and External refers to User processes
C. Both a) and b)
D. Internal fragmentation frees memory within the memory allocated to a process, that cannot be used to satisfy requests of other processes but External fragmentation the MMU might free memory that are fragmented into many non-contiguous holes and makes it available to relocate whole processes in physical memory <u>Rétt!</u>

Fyrirlestraræfing 11

1. Demand PMMU include "Page Faults". What procedure do many of today OS use to reduce overhead of swapping out- and in pages?

A. To reduce overhead of swapping- out and in pages OS use Demand Paging and no page fault will interrupts program execution <u>Rétt!!</u>
B. They check the PCB table to determine whether page is valid/invalid and make a list of idle frames stored in a list within Main memory
C. Both a) and b)
D. None of above

2. Given reference string: 1, 3, 2, 1, 1, 2, 3, 4. What number of frames will provoke 4 page faults?

A. 1 frames are needed
B. 2 frames are needed
C. 3 frames are needed <u>Rétt!</u>
D. 4 frames are needed

3. What is- and how is the modified bit controlled?

A. The modified bit is related to an OS system user command and indicates the status of the PCB (modified or not modified)
B. Modified/dirty bit for a page is set by HW whenever any byte in the page is written into, indicating that the page has been modified. When a page for replacement is selected the modify bit is examined and if it is set than the related page has been modified from the time it was read in from the disk <u>Rétt!</u>
C. The modified/dirty bit for a page is set by OS SW whenever any byte in the page is read, indicating that the page has been read. When a page for replacement is selected the modify bit is examined and if it is set than the related page has not been modified from the time it was read in from the disk
D. None of above

4. How does chosen Demand PMMU algorithm handle information from the modified bit?

A. It checks the process status within the Ready/Busy waiting queue and updates the PCB accordingly if the process is in front of the Ready queue
B. It keeps track of the page table entry and uses that info to update the table <u>Rétt!</u>
C. It initiates a related interrupt (mod-bit=set/not set) to the CPU. Microprograms will take care of tracking the page table entry and uses that info to reserve disk space on secondary storage
D. None of above

5. What are the main problems in many of today OS when using the Demand PMMU Least Recently Used algorithm?

A. There are limitation to Main memory storage in most of today computers
B. The PMMU found in a most modern CPU do not support timestamping a page at each access and are only implementable with special extra hardware Rétt!
C. It is much slower due to many calculations than other DPMMU policy algorithm and slows down the performance
D. Both a) and c)