Operating Systems 20234 Spring Midterm Exam1 (2024/03/13)

1. (6%) If it takes 20ns to search the TLB and 120ns to access memory, how long is the effective-memory-access time for a 90% hit ratio?

 Ans. The effective average memory access time is

(20 + 120) * 0.9 + (20 + 120 + 120) * 0.1 = 152ns

2. (6%) Given a disk with 200 tracks, numbered from 0 to 199, Assume the disk head is initially located at track 120 and was moving in the direction of increasing track number. Let the requested tracks, in the order received by the disk scheduler, be 55,58, 39, 18, 90, 160, 150, 38, and 186. What is the order that the requests are serviced by using the SCAN, C-SCAN, and LOOK scheduling algorithms, respectively?

Ans.

SCAN: 150, 160, 186, 90, 58, 55, 39, 38, 18

C-SCAN: 150, 160, 186, 18, 38, 39, 55, 58, 90

LOOK: 150, 160, 186, 90, 58, 55, 39, 38, 18

3. (5%) How many processes does the following code segment create including the original process?

```
void main() {
  for ( int i = 0 ; i < 3 ; i++ ) {
    if ( fork() == 0 ) {
      fork() ;
    } // if
  } // for
} // main()
```

Ans. 27

4. (3%) What is Belady's Anomaly?

Ans. Belady's Anomaly refers to the counterintuitive situation in operating system memory management where increasing the number of page frames results in an increase in the number of page faults.

5. (3%) How can a processor running in user mode be switched to kernel mode?

Ans.

A processor running in user mode can switch to kernel mode by executing a trap instruction, also known as a software interrupt.

(Some processors also offer explicit system call instructions. This is a faster mechanism since it does not need to read the branch address from an interrupt vector table (in memory) but keeps that address in a CPU register.)

6. (6%) What are the advantages of threads compare to processes?

Ans.

Threads are more efficient. Much less overhead to create: no need to create new copy of memory space, file descriptors, etc.

Sharing memory is easy (automatic.) No need to figure out inter-process communication mechanisms.

Take advantage of multiple CPUs or multiple cores. Program can scale with increasing # of CPUs.

7. (6%) What is the concept of virtual memory management?

Ans.

Virtual memory is a technique that separates user logical address space from physical memory. When executing a program, only part of logical address space needs to be in physical memory for execution. Logical address space can therefore be much larger than physical address space.

Allows physical memory to be shared by several processes. Allows using less memory for more efficient process creation.

8. (6%) Please list at three commonly used criteria for CPU scheduling.

Ans.

CPU utilization; Throughput; Turnaround time; Waiting time

Response time; Time slice (or time quantum)

9. (4%) What is the problem with priority scheduling? What is the solution?

Ans.

(1) Low priority threads may never execute. (starvation)

(2) As time progresses increase the priority of each thread. (aging)

10. (6%) Please briefly describe the Earliest Deadline First (EDF) and the Least Slack Time First (LSTF) scheduling algorithms? When there is not enough time for everything, what is the possible phenomenon for the two scheduling?

Ans.

EDF: The process with earliest deadline would be run first, and preemption happens when a process with earlier deadline arrives.

LSTF: The process with the smallest slack (deadline – remaining computation) would be run first, and preemption happens when a process with smaller slack time arrives.

If there is no enough time, EDF may finish some tasks, while LSF may miss all the deadline by approximate amount.

11. (6%) What are the internal and external fragmentations?

Ans.

Internal fragmentation occurs when a process or program is allocated more memory than it needs, and the unused portion of the memory remains allocated and unusable. This can occur when memory is allocated in fixed-size blocks, and the requested size of the memory allocation is smaller than the block size. The unused portion of the memory in the block is called internal fragmentation.

External fragmentation occurs when there is enough total memory available to satisfy a memory allocation request, but the memory is not contiguous or available in a single block. This can happen when memory is allocated and deallocated frequently, leaving small holes or gaps in the memory space that are too small to be used for a new allocation. These gaps can accumulate over time, resulting in a situation where there is enough free memory, but it is scattered in small chunks across the memory space, making it difficult to find a contiguous block of memory to satisfy a large memory allocation request.

12. (3%) What are the three typical types of i/o devices? Give an example of each device.

Ans.

Character device – keyboard

Block device – flash memory

Network device – Ethernet

13. (3%) Briefly describe the concept of "thread", "process" and "program".

Ans.

A thread can be thought of as the part of a process that is related to the execution flow.

A program is a file that contains of code and static data.

A process is a program in execution.

14. (8%) Please describe when priority inversion occurs? How to solve it?

Ans.

(a) it can occur in a multi-tasking environment when a higher-priority task is waiting for a resource (such as a lock or a semaphore) that is currently held by a lower-priority task.

(b) Priority Inheritance may be used to avoid priority inversion. It increases the priority of a thread that is holding a lock to the maximum priority of all waiting threads for the associated lock. When the lock is released, the priority goes back to its normal level.

15. (3%) Given a UNIX i-node with ten direct blocks, one single indirect block, one double indirect block and one triple indirect block. Assume that the size of a block and the block address are 16Kbytes and 4 bytes, respectively. What would be the size of the largest file allowed in byte?

Ans.

Each indirect block can record 16kbytes∕4bytes=4k block addresses

Largest file size =

$(10 + 4k + (4k)^2 + (4k)^3)$ blocks × 16k bytes/block≅1024 TB

16. (6%) Please give at least 3 differences between user-level thread and kernel-level thread.

Ans.

User-level threads and kernel-level threads are two different approaches to implement threading in an operating system. The key differences between them are as follows:

1. Management: User-level threads are managed entirely by the application, whereas kernel-level threads are managed by the operating system.

2. Scheduling: User-level threads are scheduled by the thread library or the application, while kernel-level threads are scheduled by the operating system's scheduler.

3. Context Switching: Switching between user-level threads is faster because it doesn't require the intervention of the operating system. In contrast, kernel-level threads require a context switch to move from user mode to kernel mode, which can take more time.

4. Scalability: Kernel-level threads can take advantage of multi-core processors and can run concurrently on different cores. In contrast, user-level threads are bound to a single processor, which limits scalability.

5. Robustness: Kernel-level threads are more robust as they can survive in the face of a thread failure. In contrast, a failure of a user-level thread can bring down the entire process.

17. (2%) What is the race condition?

Ans.

A race condition occurs when two or more threads or processes access a shared resource such as a variable, file, or database record, and the final outcome of the program depends on the order in which the threads or processes execute.

18. (2%) The File Allocation Table of Microsoft's FAT32 file system is a variation of:

(A) contiguous allocation.

(B) indexed allocation.

(C) linked allocation.

(D) combined indexing.

Ans. (C)

19. (2%) Which component of a process is shared across threads?

(A) CPU Registers

(B) Global variables

(C) Stack pointer

(D) program counter

Ans. (B)

20. (2%) A process that has terminated, but whose parent has not yet called wait(), is known as a _____ process.

(A) zombie

(B) orphan

(C) terminated

(D) init

Ans. (A)

21. (2%) The memory map contains several regions, which of the following statements is true?

(A) Stack can grow via a system call that requests more memory.

(B) Text (code) & initialized data come from the stored program.

(C) Heap expanded automatically.

(D) none of above.

Ans. (B)

22. (2%) Page fault frequency monitoring is a technique for:

(A) evaluating the effectiveness of the MMU.

(B) evaluating the effectiveness of the TLB.

(C) optimizing the interrupt response time for page faults.

(D) managing the resident set for all processes in the system.

Ans. (D)

23. (2%) Thrashing occurs when:

(A) The sum of the working sets of all processes exceeds available memory.

(B) The scheduler flip-flops between two processes, leading to the starvation of others.

(C) Two or more processes compete for the same region of shared memory and wait on mutex locks.

(D) Multiple processes execute in the same address space.

Ans. (A)

24. (2%) Large page sizes increase:

(A) The working set size.

(B) Internal fragmentation.

(C) The page table size.

(D) External fragmentation.

Ans. (B)

25. (2%) A semaphore puts a thread to sleep:

(A) if it tries to decrement the semaphore's value below 0.

(B) if it increments the semaphore's value above 0.

(C) until another thread issues a notify on the semaphore.

(D) until the semaphore's value reaches a specific number.

Ans. (A)

26. (2%) There are three processes on a system with the following performance requirements:

process A runs every 50 milliseconds, using 30 milliseconds of CPU time.

process B runs every 60 milliseconds, using 20 milliseconds of CPU time

process C runs every 500 milliseconds, using 10 milliseconds of CPU time

Use rate-monotonic analysis to assign priorities to the three processes. Assume that priorities range from 0 to 90, with 0 being the highest. Which of the following is a valid set of priority assignments?

(A) PA=40, PB=30, PC=100

(B) PA=50, PB=40, PC=90

(C) PA=70, PB=50, PC=60

(D) PA=40, PB=60, PC=80

Ans. (D)

27. (2%) Compared to a non-preemptive scheduler, a preemptive scheduler can move processes from the:

(A) running to the blocked state.

(B) ready to the running state.

(C) running to the ready state.

(D) blocked to the ready state.

Ans. (C)

28. (2%) A system has 32-bit logical addresses and 1 gigabyte main memory with page size of 32 Kbytes. How many entries of an inverted page table would contain:

(A) 1K entries

(B) 4K entries.

(C) 16K entries.

(D) 32K entries.

Ans. (D)