

银行家算法 + 死锁算法

- 银行家算法无法确定死锁

Rotation Latency

Semaphore

True or False

1. Aging is an effective solution to priority inversion
 - F
2. Dual-mode operation is supported by the CPU to protect the OS from applications
 - T
3. TLB works well because of spatial locality and temporal locality, where temporal locality means accesses to the same page tend to be close in time
 - T
4. Peterson's solution is an example of the spin-based locks
 - T
5. TLB entries can be extended with ASID to and TLB flushes during context switch
 - T
6. In Linux, kernel logic addresses are physically continuous; most kernel data structures, like page tables or per-process kernel stacks, are stored in kernel logic address
 - T
7. Memory-mapped I/O does not need spatial I/O instructions to operate the I/O ports and registers
 - T
8. SJF can be regarded as a dynamic priority scheduling where its priority is the next CPU burst time
 - F
9. A priority scheduling algorithm must be preemptive
 - F
10. Exit() is always invoked when the process terminates
 - T
11. seL4 is a formally verified micro kernel
 - T

12. A process switches from running to waiting state when it has used up its CPU quota and begin to wait until its next turn
 - F
13. Copy-on-write allows the parent process and the child process to share memory, so race condition may occur if they write to the same global variable concurrently
 - F
14. Peterson's solution can be extended to more than two processes with some modification
 - T
15. When CPU scheduling is considered, a process can be described as either I/O bound or memory bound
 - F
16. SJF scheduling algorithm is optimal with respect to the average turnaround time.
 - T?
17. In Linux completely fair scheduler, a process with higher priority has higher decay factor for its virtual run time
 - T
18. Bounded waiting requires a process trying to enter the critical section will eventually get in if no process is currently in it
 - T
19. Page table pages cannot be swap out to the disk by the Linux kernel
 - F
20. IA-32 only supports 6 segments per process as it only has 6 segment registers
 - F?
21. The inverted page table is used to translate physical address into virtual addresses
 - T
22. DMA controllers can steal memory access cycles from the main CPU
 - F
23. The memory management unit (MMU) is a software component of the operating system to translate logical addresses to physical addresses
 - F
24. When a user process accesses a virtual address, it is the operating system that converts the virtual address into a physical address
 - F
25. With paging, virtually continuous memory can be physically discontinuous
 - T

26. Both segmentation and paging schemes are used in IA-32
- T
27. A round-robin scheduling algorithm is non-preemptive, because it does not preempt the running process when another process switches from waiting to ready
- F
28. Re-parenting happens when the parent process terminates before the child process
- T

Short Answers

1. A solution to the critical section problem must satisfy mutual exclusion, bounded waiting and progress.
2. The SV39 mechanism (some introductions). What are the page size of a Gigapage, Megapage and page?
 - Gigapage $2^{12+18} = 2^{30}$
 - Megapage $2^{12+9} = 2^{21}$
 - Page: 2^{12}
3. The limitations of the Base & Bounds scheme are internal fragmentation, cannot support larger address space, hard to do intercrossing sharing
4. What is TLB? When does uCore need to refresh TLB?
 - TLB: Translation Lookaside Buffer, acts as a cache, allows fast address translation
5. The limitations of the segmentation scheme are OS must save and restore all pairs of segment registers, segment may grow, management of free space of physical memory with variable size segment, external fragmentation
6. What are the 3 privilege levels of RISC-V?
 - User mode, Supervisor mode and Machine mode
7. The 3 general methods used to pass system call parameters are registers, blocks, stacks
8. Hard/Soft link in iNode-based file systems are different. When creating a hard/soft link? when deleting a hard/soft link?
 - create hard link: create a directory entry pointing to the iNode of existing file
 - delete hard link: decrease the link count, when link count = 0, the data block and the iNode will be de-allocated
 - creat soft link: create a new iNode, whoe targe path name are stored using the space of block pointers
 - delete: target remain unchanged

9. The FAT32 file system stores file names into directory entry and file attributes in directory entry, but ext2/3 file system stores file names in directory entry and file attributes in iNode.
10. What are the 3 thread models for mapping user-level thread to kernel-level thread?
- one-to-one mapping, many-to-one mapping, many-to-many mapping
11. The memory contents are shown as follows, what is the value of an integer at address 0x20 if the CPU is little endian(in hexadecimal)?

Address	Content
0x20	0xe7
0x21	0x34
0x22	0x42
0x23	0xdd

- 0x dd 42 34 e7
12. Belady is the phenomenon in which increasing the number of page frames results in an increase in the number of page faults for certain memory access patterns. This phenomenon is commonly experienced when using the FIFO page replacement algorithm.

Question and Answers

Q1

On some CPU architecture, the page size is 16B, each page table entry is 4B, the virtual address has 8 bits.

- (1) What is the size of the virtual space?
- (2) If each level of page table fits into one page, how many levels of page tables are needed?
- (3) If the relevant last-level page table is shown as below, what is the physical address of 0x64?

index	PFN
00	0x15
01	0x24
10	0x3e
11	0x64

page size: $16\text{B} = 2^4 \text{B} \Rightarrow 4 \text{ bits offset}$

PTE size: 4B

virtual address: 8 bits

(1) The size of the virtual space is 2^8

(2)

the page table fits into one page, therefore there are $16B/4B = 4$ PTEs/page

there are 2 levels of page tables are needed

(3)

virtual address = $0x64 = 0110\ 0100$

the L1 page table index is 01, the L2 page table index is 10, the offset is 0100

therefore the physical address is $0x3e4$

Q2

Suppose we have 3 page frames(1, 2, 3), 4 virtual pages(A, B, C, D), consider the following reference string: A B C A D B A C B C. What are the number of page faults for the following page replacement policies? Show each step in the table to get full points.

(1) MIN

(2) LRU

(3) FIFO

(4) Clock (hint: only second page reference set reference bit)

MIN

	A	B	A	C	A	D	B	A	C	B	C
1	A	A	A	A	A	A	A	A	C	C	C
2		B	B	B	B	B	B	B	B	B	B
3				C	C	D	D	D	D	D	D

LRU

	A	B	A	C	A	D	B	A	C	B	C
1	A	A	A	A	A	A	A	A	A	A	A
2		B	B	B	B	D	D	D	C	C	C
3				C	C	C	B	B	B	B	B

FIFO

	A	B	A	C	A	D	B	A	C	B	C
1	A	A	A	A	A	D	D	D	D	D	C
2		B	B	B	B	B	B	A	A	A	A
3				C	C	C	C	C	C	B	B

CLOCK

	A	B	A	C	A	D	B	A	C	B	C
1	A	A	A	A	A	A	A	A	A	A	A
2		B	B	B	B	D	D	D	C	C	C
3				C	C	C	B	B	B	B	B

Q3

In a computer system that supports demand-paging, the memory access time is 50 ns, the average page fault service time is 5 ms. The page fault rate p is 1/5000. What is the Effective Access Time?

$$\text{EAT} = \left(1 - \frac{1}{5000}\right) \times 50\text{ns} + \frac{1}{5000} \times 5 \times 10^6\text{ns} \approx 50 + 1000 = 1050\text{ns}$$

Q4

Consider a set of 4 processes, with their arrival time and CPU burst time shown in the table. What is the average turnaround time with the following CPU scheduling algorithm?

Process ID	Arrival time t	Estimated burst time
P1	0	6
P2	1	3
P3	2	1
P4	4	2

(1) Non-preemptive SJF

(2) Preemptive SJF

(3) Round-robin with a scheduling quantum of 2 (let's assume here if two process arrive at the ready queue at the same time, the one with running-ready state transition is placed at the end of the queue).

(1) 6.75

(2) 5

(3) 7.25

Q5

In a disk scheduling algorithm, the range of the cylinders is 0~99.

The arm head is originally located at track 50.

The queue of requests is 25, 98, 52, 67, 42, 8.

What is head movement distance, considering the following disk scheduling algorithms?

Hint: for SCAN/C-SCAN/LOOK/C_LOOK, the arm **head moves towards 99 first**; for C-LOOK/C-SCAN, both directions count towards movement distance.

- (1) SSTF
- (2) C-LOOK
- (3) SCAN
- (4) FIFO

- SSTF: (50) -> 52 -> 42 -> 25 -> 8 -> 67 -> 98
 - $2 + 10 + 17 + 17 + 59 + 31 = 136$
- C-LOOK: (50) -> 52 -> 67 -> 98 -> 8 -> 25 -> 42
 - $48 + 90 + 34 = 172$
- SCAN: (50) -> 52 -> 67 -> 98 -> 42 -> 25 -> 8
 - $49 + 91 = 140$
- FIFO: (50) -> 25 -> 98 -> 52 -> 67 -> 42 -> 8
 - $25 + 73 + 46 + 15 + 25 + 34 = 218$

Q6

A Turing Class student designed an iNode-based file system, called TCFS. The structure of an iNode in Site FS has 256 bytes, with 10 direct block pointers, 2 single indirect block pointers, and 1 double indirect block pointer. A block pointer takes 8 bytes. The block size is 8 KB.

- (1) What is the maximum size of a file in TCFS?
- (2) What is the maximum size of a file system with TCFS format?

iNode size: $256 \text{ B} = 2^8 \text{ B}$

- 10 direct block pointers
- 2 single indirect block pointers
- 1 double indirect block pointers

block size: $8\text{KB} = 2^{13} \text{ B}$

block pointer: $8 \text{ B} = 2^3 \text{ B}$

contains: $2^{13} / 2^3 = 2^{10}$ addresses

(1) The maximum size of a file in TCFS is

- $10 \times 2^{13} + 2 \times 2^{10} \times 2^{13} + 1 \times (2^{10})^2 \times 2^{13}$

(2)

- block size: $8\text{KB} = 2^{13} \text{ B}$
- block pointer: $8 \text{ B} = 64\text{-bit} = \text{block address length}$
- The maximum size of a file system is $2^{13} \times 2^{64} = 2^{77}$

