Brian Wortz

CSC 345-02

Operating Systems

Spring 21

Brian Wood | Final Exam

I.

a) System Call: The Programming Interface to the services provided by the OS

Ex. fork()

System Program: Convenient Environments for Program development & execution

Ex. file management software

b) Interrupt: Priority signal that immediately changes the current process from the running to the ready state and the priority process to the running state

DSR: Examines & determines how to handle interrupt after being invoked by interrupt.

11

a) fork(): Creates a child process

exec(): Replace current process with new process

Wait(): Suspend current process until child has exited

exit(): Terminate process

b) Pipe: Conduit allowing processes to communicate

Anon/Name/Unnamed Pipe: Named can be accessed w/o a parent child relationship. Also Bidirectional

Unnamed – Can't be accessed outside of process that created it. Typically Parent creates Pipe to communicate with Child. Unidirectional

Anonymous – Same as unnamed/ordinary for Windows

III

A semaphore is a shared resource protected by a mutex so that it can only be accessed by one thread at a time.

Semaphore provides more sophisticated ways for process to synchronize.

Use semaphore rather than Mutex when one thread is waiting for another. Mutex is used when one thread is waiting on something in the same thread

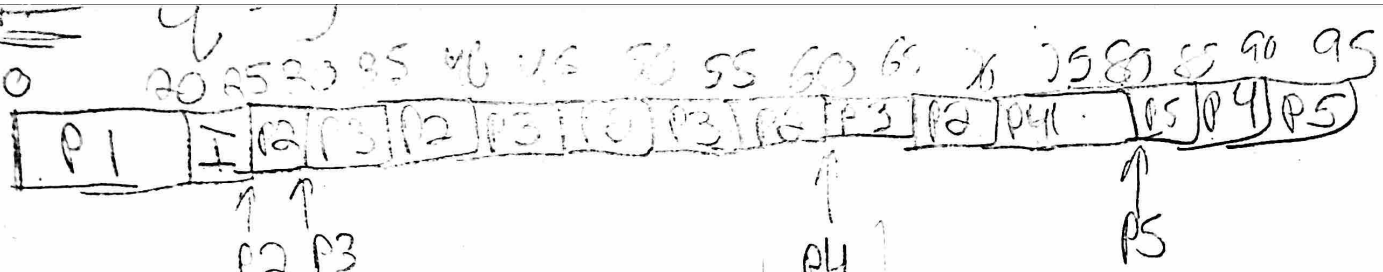
III-2

- Mutual Exclusion
- Hold & Wait
- No Preemption
- Circular Wait

III-3

Resource Allocation Graph: Analyze # of Resources and # Resources requested by Processes. Look for cycles and # instances of resources per Resource type

Banker's: Made up of safety algorithm and resource Request Algorithm. Use a need matrix to determine if sufficient resources



P1: ~~20~~ 0

P2: ~~25~~ ~~20~~ ~~15~~ ~~10~~ ~~5~~ 0

P3: ~~30~~ ~~25~~ ~~10~~ ~~5~~ 0

P4: ~~15~~ ~~5~~ 0

P5: ~~10~~ ~~5~~ 0

Turnaround

P1: 20 - 0 = 20

P2: 70 - 25 = 45

P3: 65 - 30 = 35

P4: 90 - 60 = 30

P5: 95 - 80 = 15

P5 Shortest Turnaround

Wait

P1: 0

P2: 45 - 25 = 20

P3: 35 - 20 = 15

P4: 30 - 15 = 15

P5: 15 - 16 = 5

P2 Largest Wait

CPU Utilization: $\frac{90}{95} = 94.7\%$

IV-4

Can Result in Starvation: SJF, Priority

V

a) Contiguous Allocation: Allocates a single section of memory to a process or file

b) Segmentation: Noncontiguous allocation of variable sized partitions where program is a collection of segments

c) Paging: Physical Address Space is non-contiguous. Process allocated space where available.

V-2	Contiguous	Segmentation	Paging
Internal	✓	✓	✓
External	✓	✓	

Inverted Page Table: Tracks all physical pages so one entry for each real page of memory

merits: Decrease memory needed to store each Page table

VI 7 2 3 1 2 5 7 7 0

LRU

7	2	3	1	2	5	7	7	0
7	7	7	1	1	1	7	7	
	2	2	2	2	2	2	2	
		3	3	3	5	5	5	
F	F	F	F		F	F		F

LRU: 7 faults

Optimal

7	2	3	1	2	5	7	7	0
7	7	7	7	7	7	7	7	7
	2	2	2	2	5	5	5	5
		3	1	1	1	1	1	0
F	F	F	F		F			F

Optimal: 6 faults

VI-2

a) N
b) N
c) N
d) Yes
e) Yes

VI-3

Demand Paging: Bring a Page into memory only when it is needed

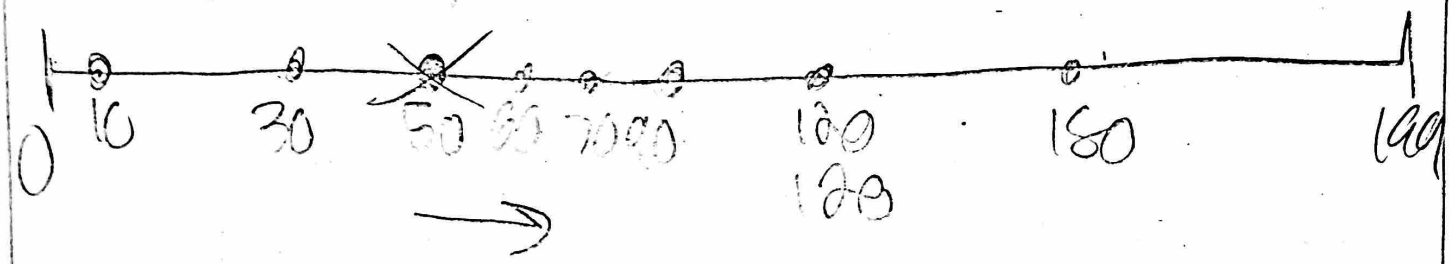
Page Fault: Requested Page is not currently in RAM, needs to be brought in

VI-4

Second Chance - FIFO Replacement Algorithm. Before Replacement, the reference bit is inspected. If it is 1, it is given a second chance and not cleared but its reference bit is set to 0. If it is 0 it gets replaced initially,

VII

$P_n = 50$, L → R



Done in Excel

SCAN: 338

C-SCAN: 378

LOOK: 300

C-Look: 320

VII-3

RAID - RAID stores data redundantly. This greatly improves reliability since there are copies of everything

RAID 0 - Non Redundant striping

RAID 1 - Mirrored Disks

RAID 10 - Disk Mirroring & Striping

VDF-1 10 Direct Blocks

4KB Pointers/Block

4B Addresses

1KB Pointers/Block

$10 \cdot 4K = 10 \cdot 4 \cdot 1024 = 40960\ B$

$40960 + 1024 \cdot 4 \cdot 1024 + 1024 \cdot 1024 \cdot 4 \cdot 1024 + 1024 \cdot 1024 \cdot 1024 \cdot 4 \cdot 1024$

$4.402 \times 10^{12} = 4.100\ GB$

VDF-5

~~Bit Map~~

Linked/Free List - Easy to get Contiguous Space W/ No wasted Space & No need to traverse whole list if # free blocks recorded

VFAT-C

Pros: Has no external fragmentation and file can grow as long as there are free blocks available as opposed to I-Node which is limited by # of pointers index block can hold

Cons: Linked List is less efficient to traverse and I-Node allows direct access

D+

a) Mouse

Buffering: Y
Spacing: N
Caching: N
I/O: Interrupt Driven

b) Network

Buffering: N
Spacing: N
Caching: Y
I/O: Polled

IX-2

Memory Mapped I/O

- Eliminates the need for special I/O from the instruction set so doesn't need enforcement of protection Rules for user programs executing these instructions

Device Control Register: Drivers place commands, addresses, and data to write/read in the Device Control Register

X

a) Type 0: Hardware based solution that provide support for VM creation & management via firmware

Type 1(1): OS like software built to provide virtualization

Type 1(2): General Purpose OS that provide standard functions as well as VMM functions

Type 2: Applications that run on standard OS but provide VMM features to guest OS

X-2

Live Migration - Move VM from one host to another

How - Source VMM establishes connection w/ target, target creates guest, source sends all read-only & read-write pages & make all clean, then VMM tells target to start running after checking state.

Why - No interruption of user access

X-3

Networked OS: Users aware of multiplicity of machines, Access to resources of various machines done explicitly.

Distributed: Users not aware of multiplicity of machines

Networked OS - Transfer Data across systems via FTP after establishing an explicit connection. Not user friendly.

Distributed OS - Much more user friendly since users not aware of multiplicity of machines. Can transfer data of computation across systems

X-4

a) Robustness

Need a fault tolerant system using fault detection reconfiguration, and recovery

b) Transparency (can the system be transparent)

System should appear as a conventional, centralized system to the user by using things like user mobility

c) Scalability (can the system be scalable)

The system can accomplish this by data compression or deduplication to cut down on storage which will allow it to react gracefully to increased load

X-5

No Problems ☺