Department of Computer Science & Engineering
Indian Institute of Technology Bhilai
CS250 – OPERATING SYSTEMS

End - Semester Examination
April 26, 2024

Time: 2 hours | Maximum Marks: 100

Student's Name | Roll No.

- Instructions
  - Answer in the space provided and show steps whenever applicable.
  - Use rough sheet for supplementary work and attach it with this answer script

1. (a) Consider a system with a 6 bit virtual address space, and 16 byte pages and frames. The mapping from virtual page numbers to physical frame numbers of a process is [$4 + 4 = 8$]

$$(0, 8), (1, 3), (2, 11), (3, 1)$$

Translate the following virtual addresses to physical addresses. Note that all addresses are in decimal.

- 20
- 40

(b) How many memory references does the instruction on line number 3 have in a segmentation and paging based virtual memory system. Justify.

```
1 void func() {
2   int x = 3000;
3   x = x + 3;    // How many memory references here?
4   ...
```

Segmentation | Paging

2. Recall the multi-level page-table control-flow as given below. Now consider the following *linear* page table for a 16KB address space. $[5 + (2 \times 3) + 3 + 2 + 4 = 20]$

```
VPN = (VirtualAddress & VPN_MASK) >> SHIFT
(Success, TlbEntry) = TLB_Lookup(VPN)
if (Success == True)   // TLB Hit
    if (CanAccess(TlbEntry.ProtectBits) == True)
        Offset   = VirtualAddress & OFFSET_MASK
        PhysAddr = (TlbEntry.PFN << SHIFT) | Offset
        Register = AccessMemory(PhysAddr)
    else
        RaiseException(PROTECTION_FAULT)
else                  // TLB Miss
    // first, get page directory entry
    PDIndex = (VPN & PD_MASK) >> PD_SHIFT
    PDEAddr = PDBR + (PDIndex * sizeof(PDE))
    PDE     = AccessMemory(PDEAddr)
    if (PDE.Valid == False)
        RaiseException(SEGMENTATION_FAULT)
    else
        // PDE is valid: now fetch PTE from page table
        PTIndex = (VPN & PT_MASK) >> PT_SHIFT
        PTEAddr = (PDE.PFN << SHIFT) + (PTIndex * sizeof(PTE))
        PTE     = AccessMemory(PTEAddr)
        if (PTE.Valid == False)
            RaiseException(SEGMENTATION_FAULT)
        else if (CanAccess(PTE.ProtectBits) == False)
            RaiseException(PROTECTION_FAULT)
```

PFN	valid	prot	present	dirty
10	1	r-x	1	0
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
23	1	rw-	1	1
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
-	0	—	-	-
28	1	rw-	1	1
4	1	rw-	1	1

(a) Draw the address space diagram for the two-level page table corresponding to the above linear page table. Clearly show the *page directory* and the associated pages that it redirects to in the main memory. Clearly state the number of bits to be allocated to each index in the virtual address.

(b) Generate three virtual addresses and explain their translations which will lead the control-flow to hit the following lines. Refer to your diagram if required.

- Line 16
- Line 23
- Line 25

(c) If the size of each PTE is 32 bits, compare the sizes of both linear and multi-levels PTs for this example.

(d) What is the significance of the *dirty* bit in the page table?

(e) Assume that the system allows a maximum of 1024 processes to run concurrently. Calculate the maximum memory-space required to store the page tables of all processes considering both linear and multi-levels PTs approaches. Which one takes up more space? In the light of your answer justify why one would want to use a multi-levels PT instead of linear PT.

3. Consider a cache of size of **three** page frames and that a program requests pages in the following order: $[6 + 6 + 4 + 4 + 2 \times 4 = 28]$

$$a, c, b, d, a, c, e, a, c, b, d, e$$

(a) Using the FIFO page replacement algorithm do a page trace analysis (page accessed, hit/miss, page evicted, cache current state) indicating the state of the cache after each removal. Then compute the hit rate. Increase the size of cache to **four** page frames and repeat the same as above.

FIFO - Cache Size = 3

Access	H/M?	Evict	Cache State

Hit Rate =

FIFO - Cache Size = 4

Access	H/M?	Evict	Cache State

Hit Rate =

Question 3 continues on the next page...

(b) Redo the above with LRU policy.

LRU - Cache Size = 3

Access	H/M?	Evict	Cache State

Hit Rate =

LRU - Cache Size = 4

Access	H/M?	Evict	Cache State

Hit Rate =

(c) What general statement can you make from the above examples? Explain your answer.

(d) Assuming the cost of accessing memory(T_M) is around 100 nanoseconds, and the cost of accessing disk (T_D) is about 10 milliseconds calculate the average memory access time ($AMAT$) in both of the above cases.

(e) Assume that a program has **just** referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.)

- TLB miss with no page fault
- TLB miss and page fault
- TLB hit and no page fault
- TLB hit and page fault

4. (a) Recall the Dining Philosophers Problem. What are the conditions of deadlock there? Outline the two solutions with their proof. $[2+4+4+1+4+2=17]$

Question 4 continues on the next page...

(b) Recall the reusable barrier problem.

What are the initial conditions for the solution.

```
turnstile = Semaphore(__)
turnstile2 = Semaphore(__)
mutex = Semaphore(__)
```

Now complete the solution to the reusable barrier problem below:

```
# rendezvous

mutex.wait()
count += 1
if count == n :
----------------
----------------
mutex.signal()

-----------------
-----------------

# critical point

mutex.wait()
count -= 1
if count == 0:
------------------
------------------
mutex.signal()

-----------------
-----------------
```

Explain what will happen if the initial value of the Semaphore in

- `turnstile` is flipped.
- `turnstile2` is flipped.

Consider these two cases separately.

5. You have come across a new file system called Bhilai File System (`bhfs`) which has a block size of 8 KB.

   The inode for `bhfs` supports 7 direct pointers and one indirect pointer. $[2 + 2 + 2 + 5 + 4 = 15]$

   (a) Assuming 4-byte disk addresses what is the maximum file-size that `bhfs` supports.

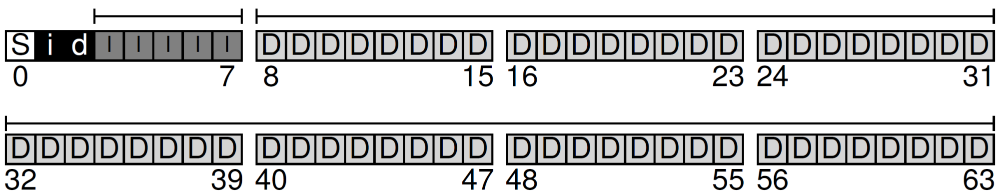
   (b) Now add one double indirect pointer to the inode. Compute the new value of the max file-size.

   (c) What decides the maximum number of files a file-system can support? Answer in terms of the file-system given in part e).

(d) Complete the following file read time-line for the file `/foo/bar`

	data bitmap	inode bitmap	root inode	foo inode	bar inode	root data	foo data	bar data [0]	bar data [1]	bar data [2]
open(bar)			**read**							
read()										
read()										
read()										

(e) Explain the following file-system organization. Assume 128 bytes per inode and 8-KB blocks.



6. Give short answers to the following: $[2 \times 3 + 1 \times 6 = 12]$

   (a) What is the difference between hard-linking and soft-linking a file.

   (b) What is a categorical mutual exclusion? Give an example.

   (c) Is swap-space the only on-disk location for swapping traffic? Explain.

   (d) Expand the following acronyms/contractions

   - PTE ________________________________
   - PDBR ______________________________
   - LRU ________________________________
   - MUTEX______________________________
   - DMA________________________________
   - inode________________________________