**Christopher Newport University**

**Department of Physics, Computer Science & Engineering**

# **CPSC 410: Operating Systems – Exam I**

**1. (5 pts) Assume the machine code equivalent of the following program is running on a processor. Disregarding page faults, please circle the statement(s) that are most likely to cause the process to be blocked.**

Int main(){

ifstream myInpuFile; -5 nothing, -4 wrong (max), -3 for lop

int j=0;

cout<<”starting”<<endl;

for (int k=0;k<1000;k++)

for (int i=0; i<1000;i++)

j+=i;

cout<<”done, total is ”<<j;

}

**2. (5 pts) What happens to a running process if all hardware interrupts are disabled on a pre-emptive Operating System? Circle all that apply.**

* It may never be swapped off the processor.
* It will be swapped off the processor at the end of its time slice. -1
* It may be swapped off the processor if it makes a call that causes a kernel mode transition.
* The operating system will poll the process to check for completion at which point a new process is loaded. -2

**3. (5 pts) Why don’t we use cooperative multitasking? Circle all that apply.**

* It requires the OS to trust programs to periodically relinquish control
* Running programs tend to make extra blocking system calls
* Process switching is much slower in cooperative than pre-emptive
* Time slicing overhead is too high
* Does not work on multi-core systems

**4. For a Round Robin scheduler, what effect does increasing the time slice have on the following. Circle one answer:**

**-2 each**

**(3 pts) throughput-** goes up goes down stays the same

**(3 pts) response time** goes up goes down stays the same

**(3 pts) number of context switches** goes up goes down stays the same

**5. (3 pts) Which scheduler has the fewest context switches? Choose 1 of the following.**

**-2**

FIFO

RR

SRTF

Multilevel Feedback Queue

Lottery

**6. (3 pts) Given the following schedulers, In general which has the worst turnaround time?**

**-2 wrong**

FIFO

RR

SRTF

**7. (10 pts) The system illustrated below has 3 levels of cache with the given access times.** **Please provide the equation to calculate the average instruction access time assuming the following conditions:**

70% of data is in T1,

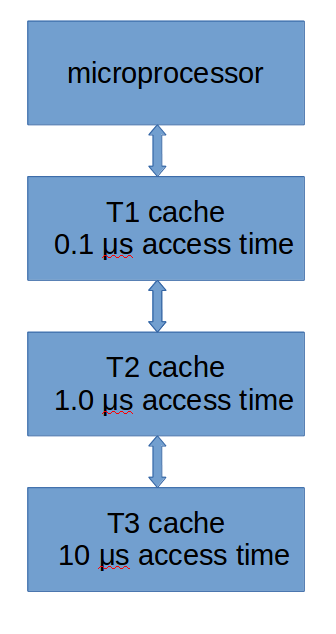
20% in T2

((.7\*0.1 + .2(.1+1.0) + .1(.1 + 1 + 10)) = 1.4

-2 div by 3

-3 did not add time to search previous cache

10% in T3

****

**8. (10 pts) A system has a 32 bit address space consisting of 8 bits of offset and 24 bits for the virtual page number. This system has a TLB with the following info:**

Valid VPN PPN ASID

1 000001 001000 00

0 010FFF F00000 01

1 001000 110000 00

0 001000 110001 01

0 0000A0 000002 01

1 001000 110001 01

Please indicate whether the following address lookup will result in a TLB miss or a hit. If a hit what is the physical address

hit address is 110000AB

**-2 dont have 110000ab**

**-3 wrong address**

**-5 tlb miss**

**ASID 00 virtual address 001000AB**

**9. A system has a 32 bit address space consisting of 10 bits of offset and 22 bits for the virtual page number.**

(**5 pts)** What is the minimum number of physical pages an array of 2048 integers will need?

Integer is 4 bytes or 2\*\*2, 2048 is 2\*\*11

2\*\*11 \*2\*\*2=2\*\*13 number bytes needed

page size is 2\*\*10

**#physical pages= 2\*\*13/2\*\*10=2\*\*8 or 8 pages**

**-1 dont account for int being 4 bytes (get 2 pages)**

**-4 get 11 pages**

**(5 pts)** Will this array be allocated contiguously in **physical memory**? Please explain why or why not?

No, pageing will allocate those 8 pages in whatever free pages are available

-4 yes, wrong

-2 no wrong

**(5 pts)**Assumming the array is allocated in the minimum number of physical pages, what will the TLB hit rate be when accessing this array in the following code? Show calculations.

**int** arr[2048];

-4 max (if something there)

-3 totally wrong

-2 wrong formula and answer

//code to load arr with values

**int** res=0;

**for**(**int** i=0;i<2048;i++){

res+=arr[i];

}

**8 misses, 2048 accesses**

**1-8/2048**

**10. (2 pts) In a paged memory system, what is the complexity (Big O notation) of the operating system allocating a page of memory?**

**O(1)**

**11. (3 pts) In a segmented memory system, what is the complexity (Big O notation) of the operating system allocating a page of memory?**

O(n)

**12. (5 pts) A program allocates 1000 bytes of memory but never deletes it. Is this memory permanently leaked when the program exits? Why or why not?**

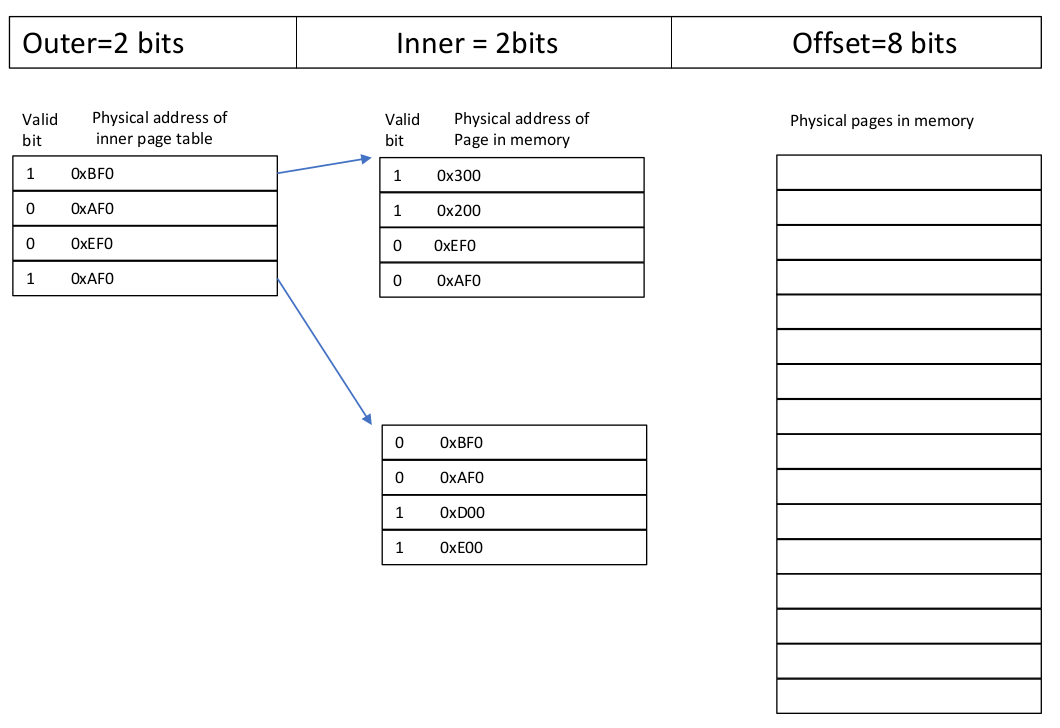
**No, OS reclaims it**

**See Chapter 14, Interlude: Memory API, just this 1/2 page "Aside: Why No Memory is Leaked Once Your Process Exits"**

**-2 nonsense answer**

**-3 yes**

**13. For a process running on the 12 bit multilevel page table system shown below**



0x311

0xD44

**(5 pts)** How many bytes are in each physical page?

-3 max

2\*\*8=256

**(5 pts)** Assuming each page table row is 2 bytes long, what is the max number of rows that a page table can have and still fit within one physical page?

2\*\*8/2\*1=2\*7=128

-2 wrong page size

-3 nonsense

**(5 pts)** How many physical pages are used by the process, excluding the page tables?

4

-2 wrong number

**(10 pts)** Please translate the following addresses. Please show where valid pages are in physical memory. Please note any invalid addresses.

-1 do not translate

-2 do not show

-1 wrong xlate (per address)

-7 max

0x011 ->00 00 0001 0001->0x311

0x833->10 00 0011 0011->INVALID

0xE44->11 10 0100 0100->0xD44