CPSC/ECE 3220 – Summer 2018 – Exam 3 **Name: _____**

No Electronics.

Write one of the words or terms from the following list into the blank appearing to the left of the appropriate definition. Note that there are more words and terms than definitions. (1 pt. each)

checkpoint	copy-on-write	core map	log	page fault frequency
page frame	page table	phase change behavior	physical address	pin
process migration	restart	segment table	self-paging	stealing a page
superpage	swapping	thrashing	TLB flush	TLB shootdown
virtual address	virtual page	working set	zero-copy I/O	zero-on-reference

1. _____ The resumption of a process from a checkpoint.

2. _____ Evicting an entire process from physical memory.

3. _____ An ordered set of steps saved to persistent storage.

4. _____ A request to another processor to remove a newly invalid TLB entry.

5. _____ Abrupt changes in a program's working set, causing bursty cache misses.

6. _____ An aligned, fixed-size chunk of physical memory that can hold a virtual page.

7. _____ The set of memory locations that a program has referenced in the recent past.

8. _____ An address that must be translated to produce an address in physical memory.

9. _____ A method for clearing memory only if the memory is used, rather than in advance.

10. _____ To bind a virtual resource to a physical resource, such as a virtual page to a physical page.

11. _____ The ability to take a running program on one system, stop its execution, and resume it on a different machine.

12. _____ A consistent snapshot of the entire state of a process, including the contents of memory and processor registers.

13. _____ A technique for transferring data across the kernel-user boundary without a memory-to-memory copy, e.g., by manipulating page table entries.

14. _____ A data structure used by the memory management system to keep track of physical page frames, such as which processes reference this page frame.

15. _____ When a cache is too small to hold its working set. In this case, most references are cache misses, yet those misses evict data that will be used in the near future.

16. _____ A resource allocation policy for allocating page frames among processes; each page replacement is taken from a page frame already assigned to the process causing the page fault.

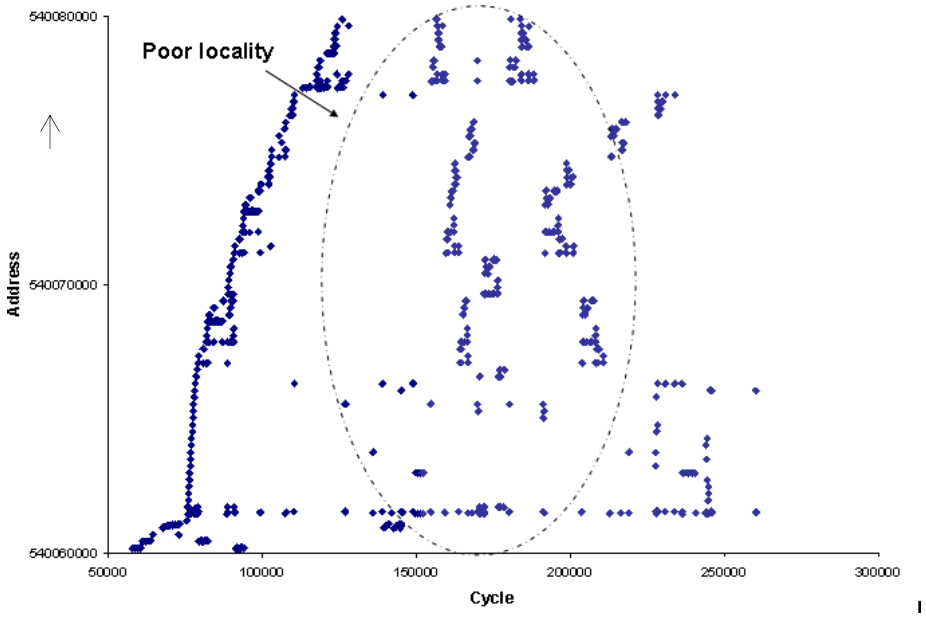
Base and Bounds / Segmentation / Paging. Circle **one or more** of BB, S, or P, as applies. (1.5 pts. each)

17. BB / S / P Can have external fragmentation.
18. BB / S / P Can be sped up by the use of a TLB.
19. BB / S / P Uses fixed-length memory allocation.
20. BB / S / P Performs a bounds check as part of address translation.
21. BB / S / P A program does not need to be stored in main memory in its entirety in order to execute.
22. BB / S / P Requires one or more CPU registers related to memory management to be loaded on process switch.

True/False. Circle either T or F. (1.5 pts. each)

23. T / F Most current operating systems use true LRU replacement.
24. T / F A TLB has one entry for each page frame in physical memory.
25. T / F Locality of reference is a property of the system's memory hierarchy.
26. T / F A TLB hit means that both read and write access is allowed to a page.
27. T / F The LRU policy replaces the page that has been least frequently used.
28. T / F A tagged TLB will flush all entries with a given tag on a process switch.
29. T / F A TLB miss requires the operating system to fetch a page table entry from disk.
30. T / F A page fault is a protection error and should cause the process to abort.
31. T / F A page or segment fault will always be immediately preceded by a TLB miss.
32. T / F Copy-on-write can be implemented on paging systems but not segmented systems.
33. T / F On a page fault, reading in a missing page from disk typically takes 10 microseconds.
34. T / F The size of an inverted page table is proportional to the size of the virtual address space.
35. T / F To prevent infinite recursion, a user-level page handler should be stored in pinned memory.
36. T / F The Zipf distribution is useful for modeling access patterns that benefit from even small caches.
37. T / F A TLB is a small address translation cache that contains copies of page or segment table entries.
38. T / F Paged memory systems require periodic memory compaction to eliminate internal fragmentation.
39. T / F A TLB shootdown is required whenever the OS adds permissions or reduces permissions for a page.
40. T / F Process eviction decisions to avoid thrashing are made by the short-term scheduler (i.e., dispatcher).
41. T / F Stack underflow and overflow can be detected easily in a base and bounds system (i.e., single segment).
42. T / F A sandbox uses hardware base and bounds registers to execute untrusted code and catch out-of-bounds jumps.
43. T / F One possible checkpointing strategy is called incremental, since copies can be made of pages modified since the last checkpoint rather than all of the pages.
44. T / F Even when every possible page in the virtual address space needs a page table entry, a multi-level page table still requires fewer total bytes to store than a single-level page table.
45. T / F FIFO page replacement can be improved by unmapping the oldest pages but placing them in a special “victim buffer” temporarily before they are replaced and overwritten. This allows “fast reclaim”.

The memory reference diagram shown below appeared in a 2007 EE Times article and is supposed to show poor locality.



46. Define “locality”. (3 pts.)

47. What is “poor” about the locality in the oval region? (3 pts.)

48. Why wasn’t the area of the diagram from 60,000-120,000 cycles included in the region? Your answer should identify the type of locality that the initial part of the diagram illustrates. (3 pts.)

A virtual address in the MULTICS system was structured as:

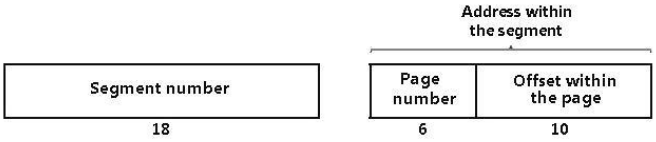


Figure 3. A 34-bit MULTICS virtual address.

Answer questions 49-50 using powers of 2. Use words as the addressable units, and do not convert to bytes.

49. Consider a data structure of 512 Ki words. How many segments do you need to store this structure? (3 pts.)

50. How many pages do you need to store the data structure in question 49? (3 pts.)

Consider a simple architecture with paged segments in which three-decimal-digit virtual addresses are used. The segment id is the first decimal digit, the virtual page number is the second decimal digit, and the offset is the last decimal digit. The physical addresses are two decimal digits: the page frame is the first decimal digit, and the offset is the second decimal digit. Give the physical address if main memory will indeed be accessed. (3 pts. each)

Segment Table	**Page Table A**	**Page Table B**
0 = points to page table A, segment length is 4 pages	0 = page frame 4, execute-only	0 = page frame 7, read/write
1 = points to page table B, segment length is 3 pages	1 = not present, execute-only	1 = page frame 5, read/write
X (other values are invalid)	2 = page frame 0, read-only	2 = not present, read/write
	3 = page frame 9, read/write	X (other values are invalid)
	X (other values are invalid)	

51. What happens if a load instruction in the process attempts a read of virtual address 024?

52. What happens if a store instruction in the process attempts a write of virtual address 024?

53. What happens if a load instruction in the process attempts a read of virtual address 124?

54. Suppose a process is assigned four page frames of physical memory, where the frames are initially empty. The process then references pages in the following sequence (using letters for VPN references in the manner of the textbook). Using the diagram format of the textbook, show how the system would respond to page faults using the FIFO replacement policy. Using the diagram format of the textbook (letter means miss, and "+" means hit). (6 pts.)

	A	**B**	**C**	**D**	**B**	**E**	**C**	**F**	**B**	**G**	**A**	**C**	**E**	**F**	**B**
1															
2															
3															
4															

55. Repeat but using LRU (6 pts.)

	A	**B**	**C**	**D**	**B**	**E**	**C**	**F**	**B**	**G**	**A**	**C**	**E**	**F**	**B**
1															
2															
3															
4															

56. Determine the number of TLB misses analysis for two approaches given below to zero the matrix "a". Assume that matrix a is stored in row-major order, that a double-precision value is 8 bytes, that the i and j index variables are register-allocated rather than memory allocated, that you have a TLB with 8 entries, that the TLB starts off empty, and that pages are 16 KiB in size. (6 pts.)

```
double a[128][128];


// approach 1

for( i = 0; i < 128; i++ ){
        for( j = 0; j < 128; j++ ){
                a[i][j] = 0.0;
        }
}


// approach 2

for( j = 0; j < 128; j++ ){
        for( i = 0; i < 128; i++ ){
                a[i][j] = 0.0;
        }
}
```

57. What is the advantage in choosing a clean (or empty) page to replace instead of a dirty page? (2 pts.)

58. What mechanism does the operating system use to distinguish a clean page from a dirty page? (1 pt.)

59. Identify and explain at least two conditions under which a multi-level translation scheme can omit building lower-level page tables. (4 pts.)

60. Describe the problem and the solution when an operating system wants to take a checkpoint of a process for which a system call handler is in progress. (3 pts.)