Solutions for CS 354 Final, Spring 2014

P1(a) 10 pts

XINU: Linear cost to access data blocks from iblock structure.
UNIX: Constant cost for small files using direct pointers, logarithmic cost using indirect pointers
5 pts

"Mice and elephants" property means that in real-world file systems most files are small and a minority are very large. The UNIX indexing structure, by using direct pointers, is able to achieve constant access time for most file access which are small.
5 pts

P1(b) 10 pts

Process A wants 100 MB memory (contiguous) but RAM has only two free regions: one of size 70 MB, another of size 30 MB. Although the total free memory matches the 100 MB, it is not contiguous hence system call will fail.
5 pts

Indirection that uses a table to translate virtual addresses into physical addresses. For the above example, in the virtual space, the fragmented 70+30 MB can be made to seem contiguous.
3 pts

Since this is done for every memory reference that needs virtual to physical address translation, it must be fast and cannot be handled by kernel software.
2 pts

P1(c) 10 pts

Saving registers onto process stack, updating process table information,

etc. [2 or more items is enough]
3 pts

Saving more process state (per-process page table register values), flusing of L1, TLB, L2 cache to avoid aliasing.
4 pts

The latter is considered higher due to misses (TLB, page faults, L1, L2) that result from flushing of caches.
3 pts

P1(d) 10 pts

Tickful: A fixed tick value triggers a system timer clock interrupt periodically.
Tickless: An interval timer is programmed to trigger a clock interrupt when needed. Hence tick values are not fixed.
4 pts

Tickful: Assuming a delta list is being used, the front elements time value is decremented every tick. When it reaches 0, the front process (and any other with equal sleep value) are readied and the scheduler called.
Tickless: Suppose the time value of the front element is S. An interval timer is programmer to generate an interrupt after S time units.
4 pts

Tickless, all other things being equal, since to conserve battery power, not running a clock interrupt handler every tick (even when there is nothing to do) is potentially less wasteful.

P2(a) 12 pts

A page fault arises when a page addressed by a CPU is not found in RAM.
3 pts

The kernel initiates disk I/O (or flash memory I/O depending on persistent storage) to read the missing page into RAM. If all frames in RAM are full, a page is evicted to free up space which results in writing the content of the page to disk. Since disk I/O is slow, the current process that page faulted is context switched out and a ready process context switched in.
6 pts

If the speed gap is sufficiently small, it may be feasible for the hardware to handle page faults by waiting on the flash memory I/O to complete without context switching out the current process.
3 pts

P2(b) 12 pts

Orignal XINU: enqueue - linear, dequeue constant
Solaris TS XINU: constant
4 pts

Searching in a multi-level feedback queue is considered constant since the number of priorities is treated as being fixed, i.e., constant. Dequeueing is still constant since the front process at the highest priority level needs to be removed. Enqueueing is constant since we assume round-robin scheduling among equal priority process which means that a processed is enqueued at the end of the list which also incurs constant overhead.
8 pts

P2(c) 12 pts

The lower half of a kernel that handles interrupts kicks in.
2 pts

Top: Instructs DMA to copy how much data to where in RAM (part of kernel memory).

DMA: Copies the number of packets/data to RAM area as specified by top half.
4 pts

Bottom: Process data in kernel memory (may also first copy from temporary kernel buffer to larger kernel buffer that's shared with the upper half), then informs the upper half that data is ready. This means unblocking the process that may have been blocking and calling the scheduler.
Upper: Copies data from kernel buffer to user buffer, then returns from system call which switches process from kernel mode to user mode.
4 pts

If the upper half does not read the kernel buffer sufficiently frequently, the buffer may build up and overflow thus losing video information.
2 pts

P3(a) 12 pts

Time 1s: (A,12)
Time 2s: (A,11)
Time 4s: (B,8) -> (A,1)
Time 6s: (B,6) -> (A,1) -> (C,1)

P3(b) 12 pts

The virtual address to be translated cannot be found in the TLB cache that holds part of the process's page table.
2 pts

Locate in RAM the process's page table. Assuming the process table is locked into RAM, read the missing entry from RAM into TLB. Resume (i.e., re-execute) which will now result in TLB hit.
4 pts

Disk I/O is needed if some page tables are located on disk.
1 pts

Lock page tables into RAM.
1 pts

2-level page tables decrease the total space required by all process' page tables in RAM. Hence it reduces the need to put some page tables on disk.
3 pts

The content of the physical address translated from virtual address is accessed in L2 or RAM.
1 pts

Bonus

File caching in RAM.
6 pts

Page replacement policy in VM through global clock that mimicks LRU.
6 pts
[any two valid examples are fine]