CS2506 Summer 2012 Solutions

Question 1 (a)

The purpose of process scheduling is to order the processes that are ready to execute in such a manner that allows them to be completed with the highest level of efficiency. The processes are organized in a queue from where the scheduler selects the next one to take control of the CPU.

What process has generally the lowest priority?

Question 1 (b)

Multilevel Feedback Queues

- Uses a bitmap to quickly determine which queue is the highest priority non-empty one.
- Interactive processes that completely use their quantum are placed back into the active queue and are scheduled in a round-robin fashion.
- Compute-bound processes that completely use their quantum are moved to the expired queue.
- When the active queue becomes empty, it is swapped with the expired queue.
- Priority and time slice length are computed each time a process is placed into a queue.

Example

- Three queues:
  - Q_0 – time quantum 8 milliseconds
  - Q_1 – time quantum 16 milliseconds
  - Q_2 – FCFS
- Scheduling
  - A new job enters queue Q_0 which is served FCFS. When it gains CPU, job receives 8 milliseconds. If it does not finish in 8 milliseconds, job is moved to queue Q_1.
  - At Q_1 job is again served FCFS and receives 16 additional milliseconds. If it still does not complete, it is preempted and moved to queue Q_2.

Question 1(c)

Scheduling domains are sets of cores which share properties and scheduling policies that can be balanced against each other. Processes from a core in one scheduling domain can be moved to another to spread to load.

- Each scheduling domain contains one or more core groups, which are subsets of cores that are treated as one entity. The actual balancing of process load happens between core groups.
- As process attributes change and resources become used up, processes are moved to scheduling domains with certain policies.

Policy Examples:

- If a processor becomes idle, and its domain has the SD_BALANCED_NEWIDLE flag set, the scheduler will go looking for processes to move over from a busy processor within the domain.
- If one processor in a shared pair is running a high-priority process, and a low-priority process is trying to run on the other processor, the scheduler will actually idle the second processor for a while. In this way, the high-priority process is given better access to the shared package.

Question 2(a)

The memory addresses are physical (real) addresses. They uniquely identify more locations and create the physical address space. On the other hand, the process layout is specified in terms of virtual addresses. Their set represents the virtual address space.

- Memory pages are 2^k byte sized areas of memory. They are part of a virtual to physical address mapping technique called paging.
- If virtual memory addresses are n bits in length; there can be 2^{n-k} pages. The upper n-k bits of the virtual address point to an entry in a page table. Page table entries hold, along with other information, a page frame number, which added to the lower n bits of virtual address, make up a physical address.
- There can also be hierarchical paging. In this case, the upper bits of the virtual address are split in two. The upper bits index a page directory, whose entries in turn index various page tables. The lower bits index into this page table the same as above to get the physical address.

Question 2(b)

PTE - Page Table Entries hold 5 main pieces of information

- ***PAGE FRAME NUMBER*** – the start of a physical page frame. An offset is added to it to get physical address.
- ***PROTECTION BITS*** – Indicate protection policies such as read-only for shared memory pages.
- ***PRESENT BIT*** – A bit which indicates a valid address translation. A page fault occurs if a physical address does not exist, or the process cannot access it.
- ***MODIFIED BIT*** – A bit which indicates whether this page has been written to recently.
- ***ACCESSED BIT*** – This bit is set if this page is accessed by a process.

Example

NOT RECENTLY USED is similar to second chance, except the modified bit M of memory blocks is checked as well as the accessed bit A. The blocks with the lowest AM values are chosen for swapping.

Question 2(c)

- If memory blocks are fixed in size, the allocation process can result in waste. More memory allocated than is necessary = internal fragmentation. Memory left unallocated = external fragmentation.
- The simplest method of allocating memory is based on dividing memory into areas with fixed partitions.
- However, the flexibility of allocating memory in either large or small blocks is needed: e.g. a free block is selected and split into two parts, the first is allocated to the process and the second is returned as free space. The allocation is done in multiples of some minimum allocation unit (OS parameter). This helps to reduce external fragmentation.

Question 3(a)

The driver can be divided in two halves:

- The upper deals with user requests;
- The lower deals with the controller.
- The two threads of control use mutual exclusion techniques in order to prevent corruption of the shared queue.

Question 3(b)

- Noop scheduler merges adjacent requests; when a request addresses sectors adjacent to sectors accessed by a request already in the queue, the two can be combined into a single request.
- Complete fair queuing scheduler maintains a separate queue for each process. Requests are merged and inserted in.

Question 3(c)

RAID ensures that any single disk failures in any of our systems have no effect on any running services. Basically a disk can fail, and the system keeps running fine.

The benefits of RAID come from a technique called "striping," which splits up the stored data among the available drives. The "stripes" of data are usually a couple of megabytes large and are interleaved between the drives. The striping system also increases the mean time between failures (MTBF), when reading data. This allows more data to be read accurately in a short period of time.

- RAID level 0: Data striping – stripping at the level of blocks but without any redundancy.
- RAID level 1: Data mirroring – a drive has its data duplicated on two different drives.

Question 4(a)

- File system metadata is information about a file system and its contents.
- The file system metadata includes:
    - The total size of the system,
    - The amount of free space,
    - The date of the last mount etc.

File System Metadata provides a means to organize data expected to be retained after a program terminates by providing procedures to store, retrieve and update data as well as manage the available space on the device(s) which contain it, thus ensuring reliability is a major responsibility of a file system.

Question 4(b)

The allocation of space in a file system is done in fixed-sized blocks.

- For keeping track of free blocks, the free bitmap can be used.
- Another solution is to use free lists, embedded in the free blocks.
- Finally, a simple list stored in a free block can be used.

Question 4(c)

File systems must be accessed in an efficient manner, as a computer deals with multiple processors over a period of time, a list of request to access the disk build up. The operating system uses a disk scheduling technique to determine which request to satisfy.

- **FCFS** (First Come, First Served)
    - perform operations in order requested
    - no reordering of work queue
    - no **starvation**: every request is serviced
    - poor performance
- **SSTF** (Shortest Seek Time First)
    - after a request, go to the closest request in the work queue, regardless of direction
    - reduces total seek time compared to FCFS
    - Disadvantages
        - **starvation** is possible; stay in one area of the disk if very busy
        - switching directions slows things down
- **SCAN**
    - go from the outside to the inside servicing requests and then back from the outside to the inside servicing requests.
    - repeats this over and over.

- reduces variance compared to SSTF.