Ph.D. Qualifying Examination

Fall 2011

Operating Systems

Question I (100 points)

Two common architectures used in operating systems are hierarchical layers and microkernels.

a) (50 points) Compare their advantages and disadvantages for their use in general operating systems.
b) (50 points) Show an architecture that combines them. Explain what is gained/lost by this combination.

Ph.D. Qualifying Examination

Fall 2011

Operating Systems

Question II (100 points)

1. (25 Pts.)
What four conditions are simultaneously required for a deadlock situation ? Describe each briefly.

2. (25 Pts.)
Explain how the dirty bit is used in demand paging.

3. (20 Pts.)
What is the Translation Look-aside Buffer ?

4. (30 Pts.)
Calculate the page fault rate on a memory with 4 page frames for the Optimal and LRU page replacement algorithms, for the following sequence of page references:

1,2,3,4,5,1,3,7,1,4,5,2,3,7,6,2,5,1

Ph.D. Qualifying Examination

Spring 2010

Operating Systems

Question I (100 points)

a) (60 points) Suppose that a disk drive has 300 cylinders, numbered 0 to 299. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 15. The queue of pending requests, in FIFO order, is: 86, 147, 291, 18, 95, 151, 12, 175, 30.
Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk-scheduling algorithms?
   1. FCFS
   2. SSTF (Shortest Seek Time First)
   3. SCAN
   4. C-SCAN

b) (40 points) Given free memory blocks of 100K, 500K, 200K, 300K, and 600K (in this order), how would each of the following algorithms place processes of 212K, 417K, 112K, and 426K (in this order) in these blocks?
   1. First-fit
   2. Best-fit
   3. Next-fit
   4. Worst-fit

Which algorithm makes the most efficient use of memory?

PhD Qualifying Examination

Fall 2010
Operating Systems

Question I (100 points)

1. (80 points) Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time.

Process	Arrival Time	Burst Time
P_1	0.0	3
P_2	1.0	6
P_3	2.0	4
P_4	4.0	1
P_5	7.0	2

Compute the average turnaround time and average waiting time for each of the following scheduling algorithms. Show work.

a. FCFS
b. Non-preemptive SJF
c. Preemptive SJF
d. RR (quantum = 2)

2. (20 points) Most contemporary operating systems use spinlocks as a synchronization mechanism on multi-processor systems, but not on single-processor systems. Explain why.

PhD Qualifying Examination

Fall 2010

Operating Systems

Question II (100 points)

1. [40 Points] What are the four conditions that must hold simultaneously in order for a deadlock to occur? Briefly explain each condition.

2. [60 Points, 20 points each] Given the following sequence of tracks on a disk:
   55, 58, 39, 18, 90, 160, 150, 38, 184

   Determine the total number of tracks traversed and the average seek (track) length using the specified algorithm below. Fill in the tables to show your answers.
   - Lowest track number is 0
   - Highest track number is 200
   - Beginning track is 100

   a) Shortest-Seek-Time-First scheduling

Next track assessed									
Number of tracks traversed									

Total tracks traversed: ________________

Average seek length: ________________

   b) SCAN (direction is heading toward the highest track)

Next track assessed									
Number of tracks traversed									

Total tracks traversed: ________________

Average seek length: ________________

c) C-SCAN (direction is heading toward the highest track)

Next track assessed									
Number of tracks traversed									

Total tracks traversed: ________________

Average seek length: _________________

Ph.D. Qualifying Examination
Spring 2009
Operating Systems

Question II (100 points)

Part 1 (50 Pts.)

a. (15 Pts.)
What are the problems of duplicating all threads when implementing fork()?

b. (15 Pts.)
What is process starvation ? How can process schedulers prevent starvation?

c. (20 Pts.)
Explain how the dirty bit is used in demand paging.

Part 2 (50 Pts.)

a. (25 points)
Assume two processes (P1 and P2) know each other and want to execute code sequences A1(), B2() and A2(), B2() synchronized, as follows:

```
P1          P2
A1();              A2();            ↓ time
....               ....
B1();              B2();
```

That is: P1 does A1() and P2 does A2(), possibly in parallel, then both P1 and P2 proceed in parallel with B1() and B2(), respectively. It is important that A code does not overlap with any B code. How could you implement this synchronization? (write pseudo-code)

b. (25 Points)
How many **new** processes are created by the following code sequence? Draw the process tree and explain your answer.

```
......
fork();
fork();
fork();
```

PhD Qualifying Examination

Spring 2009

Operating Systems

Question I: (100 points)

1. (60 points) Compute the number of page faults that may occur in a demand-paging system, using the optimal page-replacement algorithm, given that there are four physical frames available with the following reference string. Show work.

   1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 3, 4, 5, 3, 4, 5, 5, 1, 2, 3

2. (40 points) Consider the following memory resource-allocation policy. Requests and releases for resources are allowed at any time. If a request for resources cannot be satisfied because the resources are not available, then all waiting processes that are blocked for resources to be released are checked. If any of these waiting processes has the desired resources, then these resources are taken away from it and are given to the requesting process. The vector of resources for which the waiting process is waiting is increased to include the resources that were taken away.

   For example, consider a system with three resource types and the vector *Available* initialized to (4,2,2). If process P_0 asks for (2,2,1), it gets them. If P_1 asks for (1,0,1), it gets them. Then, if P_0 asks for (0,0,1), it is blocked (resource not available). If P_2 now asks for (2,0,0), it gets the available one (1,0,0) and one that was allocated to P_0 (since P_0 is blocked). P_0's *Allocation* vector goes down to (1,2,1), and its *Need* vector goes up to (1,0,1).

   a) Can deadlock occur? Explain.

   b) Can indefinite blocking occur? Explain.

Ph.D. Qualifying Examination

Spring 2008

Operating Systems

Question I: (100 points)

1. (50 points) Given memory partitions of 120K, 420K, 320K, 520K, and 220K (in order), how would each of the *First-fit*, *Best-fit*, and *Worst-fit* algorithms place processes of 300K, 400K, 140K, 100K, and 340K (in order)? Which algorithm makes the most efficient use of memory? Draw a diagram(s) to show work.

2. (50 points.) Consider the following preemptive priority-scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue but not running), its priority changes at a rate α , and when it is running, its priority changes at a rate β. All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.

   a) What is the algorithm that results from $\beta > \alpha > 0$?

   b) What is the algorithm that results from $\alpha < \beta < 0$?

   Note: You must show at least one example for each of the above to support your answers.

Ph.D. Qualifying Examination

Spring 2008

Operating Systems

Question II (100 points)

1. (35 points)

Consider the following processes, their arrival times, burst duration and priority (low number means high priority):

PID	Arrival time	Burst Duration	Priority
1	0	6	5
2	2	8	3
3	6	2	4
4	10	4	2

Draw the Gantt chart and compute the average waiting time for the following scheduling algorithms:

a) FCFS

b) Shortest job first with preemption

c) Priority with preemption

d) Round-robin with a 4 unit time slice

2. (35 points)

Explain how a web server would benefit from a multithreaded design compared with a single threaded server, and why.

3. (30 points)

A computer uses demand paging virtual memory. Handling a page fault takes 4 *ms* if the replaced page is not modified and 10 *ms* if the replaced page is modified. A replaced page is modified 20% of the time. The main memory access time is 2 *ns*. What is the maximum page fault rate for an effective access time of no more than 4 *ns* ?

Ph.D. Qualifying Examination

Fall 2008

Operating Systems

Question I: (100 points)

1. (70 pts.) Consider the following snapshot of a system:

	Allocation	Max	Available	Need
	A B C D	A B C D	A B C D	A B C D
P_0	1 1 1 2	1 2 3 5	1 2 3 1	
P_1	1 1 2 1	2 3 4 1		
P_2	1 2 1 1	5 9 1 2		
P_3	2 1 1 1	4 1 2 3		
P_4	1 2 1 2	1 7 3 2		

Answer the following questions using the banker's algorithm.

a) What is the content of the matrix *Need*? Show work.
b) Is the system in a safe state? Show work.
c) If a request from process P_2 arrives for (0,2,0,1), can the request be granted immediately? Show work.

2. (30 pts.) Briefly describe a *critical section*, and the requirements for solving the *critical section* problem:

Ph.D. Qualifying Examination

Fall 2008

Operating Systems

Question II (100 points)

1. (15 Pts.)
What are the benefits of the microkernel design compared to a monolithic approach?

2. (15 Pts.)
Explain the concept of rendezvous for synchronization with message-passing.

3. (20 Pts.)
What four conditions are simultaneously required for a deadlock situation? Describe each briefly.

4. (50 Pts.)
Consider the following processes and their parameters:

PID	*Arrival Time*	*CPU Burst Duration*	*Priority*
1	0	8	5
2	0	6	4
3	4	8	2
4	12	4	3

Draw the Gantt charts and compute the process waiting times and the average waiting times for the following schedules (low priority number means high process priority):

1. SJF – no preemption
2. SJF – with preemption
3. Priority Scheduling (with preemption) and
4. RR (with time slice = 4 time units).

Ph.D. Qualifying Examination

Spring 2007

Operating Systems

Question I: (100 points)

1. (50 points) Compute the number of page faults that may occur in a demand-paging system, using the optimal page-replacement algorithm, given that there are four physical frames available with the following reference string. Show work.

   1, 2, 3, 4, 5, 3, 4, 5, 3, 4, 5, 1, 2, 3, 4, 5, 4, 5, 1, 2

2. (30 points) Consider a demand-paging system with the following time-measured utilizations:

   CPU utilization 20%
   Paging disk 90%
   Other I/O devices 5%

   Which (if any) of the following will (probably) improve CPU utilization? Explain your answer.

   a) Install a faster CPU.
   b) Install a bigger paging disk.
   c) Install more main memory.
   d) Increase the degree of multiprogramming.
   e) Increase the page size.

3. (20 points) What are the advantages and disadvantages of a file system that provides mandatory locks instead of advisory locks? The usage of the latter is left to the users' discretion.

Ph.D. Qualifying Examination

Spring 2007

Operating Systems

Question II (100 points)

a) Explain how the dirty bit is used in demand paging.

b) How does a condition variable (from a monitor) differ from a semaphore ?

c) Consider the following processes, their arrival times, burst duration and priority (low number means high priority):

PID	Arrival time	Burst Time	Priority
1	0	6	5
2	2	8	3
3	6	4	2
4	10	2	4

Draw the Gantt chart and compute the average waiting time for the following scheduling algorithms:

a) FCFS

b) Shortest job first with preemption

c) Priority with preemption

Ph.D. Qualifying Examination

Fall 2007

Operating Systems

Question II (100 points)

a) (20 points)
Explain why spinlocks are not good for single-processor systems.

b) (40 points)
Answer these questions about the rendezvous concept:
b1. Explain the concept of rendezvous.
b2. What happens with the rendezvous if the communication channel has a 3-message buffer ?
b3. What happens with the rendezvous if only the *send*() operation is synchronous ?

c) (40 points)
A memory system with pages of size 1024 bytes has 512 frames of physical memory. Please answer these questions:
c1. What is the structure of the logical memory address ? What is the size of the page table ?

c2. For a process the content of the page table is:

Page	Frame
...	...
23	32
24	0
25	43
26	510
27	23
...	

Show how logical address 25:715 (page:offset) maps to a physical address (address parts are in base 10).

c3. Is 65789_{10} a valid logical memory address. Why ?

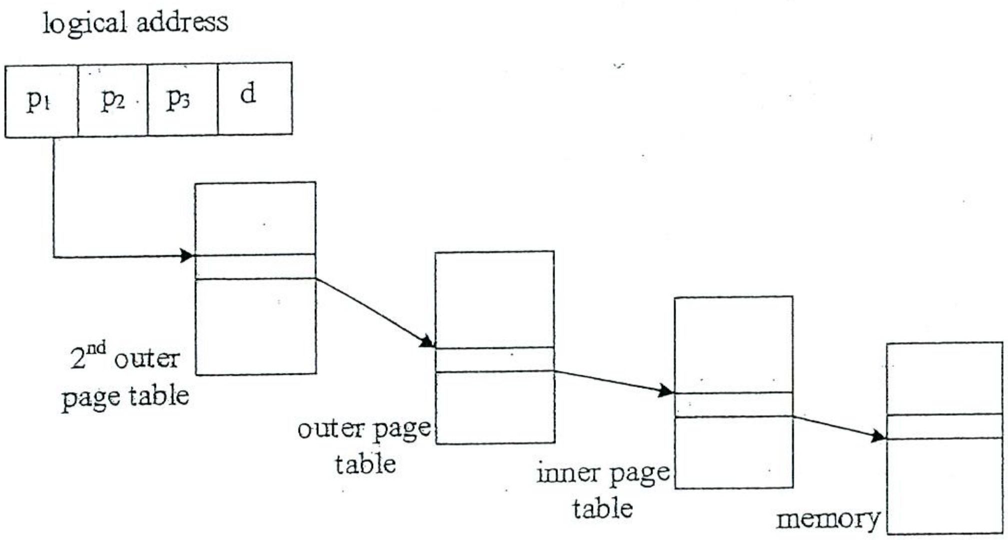
PhD Qualifying Examination

Fall 2007

Operating Systems

Question I: (100 points)

1. (70 points) Consider a paging system with a 3-level page table implementation mechanism as shown below:



   a. If a memory reference takes 300 nanoseconds, how long does a paged memory reference take?

   b. If we add associative registers, and 85 percent of all page-table references are found in the associative registers, what is the effective memory reference time? Assume that finding a page-table entry in the associative registers takes 50 nanoseconds.

2. (30 points) In memory management, segmentation and paging are sometimes combined into one scheme for good reasons. Explain!

Ph.D. Qualifying Examination
Spring 2006
Operating Systems

Question II (100 points)

1. (50 Pts.)

a. (20 Pts.)
What does the O.S. do when a process tries to access a memory page that is not present in main memory (a page fault), for a system with demand paging virtual memory ?

b. (15 Pts.)
What is the Translation Look-aside Buffer ?

c. (15 Pts.)
What are the 5 main states a process goes through during its lifetime ? Describe each state briefly.

2. (50 Pts.)
a. (15 Points)
How many **new** processes are created by the following code sequence ?

```
......
fork();
fork();
fork();
```

b. (35 points)
Assume two processes (P1 and P2) know each other and want to execute code sequences A1(), B2() and A2(), B2() synchronized, as follows:

```
P1              P2                  ↓ time
A1();           A2();
....            ....
B1();           B2();
```

That is: P1 does A1() and P2 does A2(), possibly in parallel, then both P1 and P2 proceed in parallel with B1() and B2(), respectively. It is important that A code does not overlap with any B code. How could you implement this synchronization ?

PhD Qualifying Examination

Spring 2006

Operating Systems

Question I: (100 points)

1. (70 points) Compute the number of page faults that may occur in a demand-paging system, using the following three replacement algorithms. Assume that there are four physical frames available with the reference string given below. Show work.

   {1, 2, 3, 4, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3}

   a) LRU replacement
   b) FIFO replacement
   c) Optimal replacement

2. (20 points) A computer system supporting virtual memory via paging provides its users a virtual memory space of 2^{32} bytes. The page size is 4K bytes, and the computer has 512M bytes physical memory. If a user process generates a virtual address of $12AB34CD_{hex}$, explain how the system derives the corresponding physical address.

3. (10 points) For file sharing, some OS systems provide multiple accesses to a shared file by maintaining a single copy of a file. Other OS systems maintain several copies, one for each of the users sharing the file. Discuss the pros and cons of each approach.

Ph.D. Qualifying Examination
Fall 2006
Operating Systems

Question II (100 points)

1. (a+b+c: 50 Pts.)
a. (20 Pts.)
What is process starvation ? How can process schedulers prevent starvation ?

b. (20 Pts.)
Explain how the dirty bit is used in demand paging.

c. (10 Pts.)
What is the issue with duplicating all threads when implementing fork() ?

2. (50 Pts.)
Consider the following processes with their parameters:

PID	*Arrival Time*	*CPU Burst Duration*	*Priority*
1	*0*	*8*	*5*
2	0	6	4
3	4	8	2
4	12	4	3

Draw the Gantt charts and compute the process waiting times and the average waiting times for the following schedules:
SJF – no preemption, SJF – with preemption, Priority Scheduling (with preemption) and RR (with time slice = 4 time units).
(low priority number means high process priority)

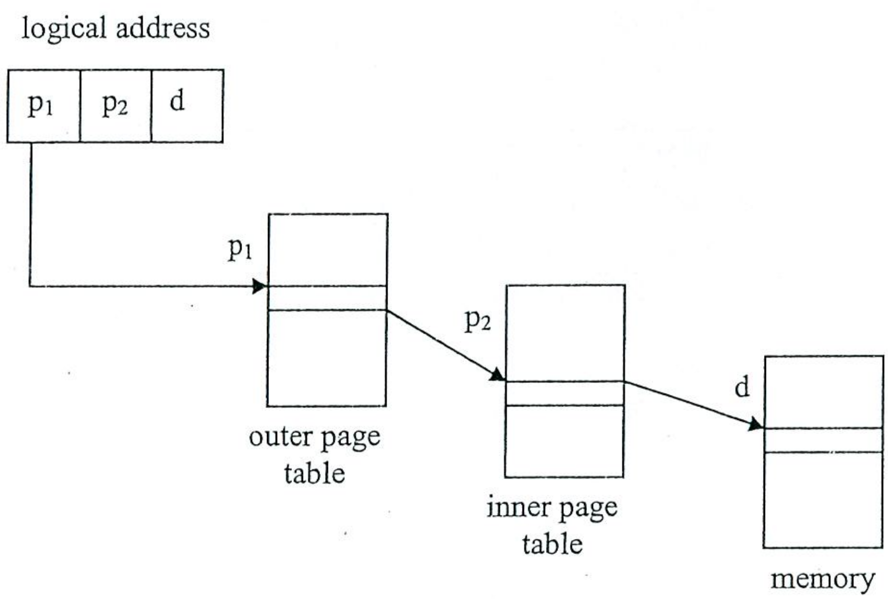
PhD Qualifying Examination

Spring 2005

Operating Systems

Question I: (100 points)

1. (60 points) Consider a paging system with a two-level page table implementation mechanism as shown below:



a. If a memory reference takes 250 nanoseconds, how long does a paged memory reference take?

b. If we add associative registers, and 80 percent of all page-table references are found in the associative registers, what is the effective memory reference time? Assume that finding a page-table entry in the associative registers takes 50 nanoseconds.

2. (40 points) Major operating systems today implement a free-block list to chain together available disk blocks that are not assigned to any file on a secondary storage disk. Space needed is allocated from the free disk blocks each time a new file is created and the pointer to the free-block list is updated accordingly. Should the pointer to the free-block list be lost due to a system crash, can the host operating system reconstruct the free-block list? Explain your answer.

Ph.D. Qualifying Examination

Spring 2005

Operating Systems

Question II (100 points)

1. (50 Pts.)

a. (15 Pts.)
What are the benefits of the microkernel design compared to a monolithic approach ?

b. (15 Pts.)
Explain the concept of rendezvous for synchronization with message-passing.

c. (20 Pts.)
What four conditions are simultaneously required for a deadlock situation ? Describe each briefly.

2. (50 Pts.)
There are *n* processes in a system. Each process executes a function `proc()` in an infinite loop. `proc()` implements a critical section that requires that at most *m* processes can execute it concurrently. The `enter()` and `leave()` functions are used to limit the number of processes within the critical section to *m*. The process code is:

```
void process() {
      while (true) {
            enter();              // limit access to m processes
            proc();               // execute critical section
            leave();              // leave critical section
            sleep(random());      // wait some time
      }
}
```

When a process calls `enter()`, it can proceed (it returns) only if there are less than *m* other processes in the critical section. The calling process will be blocked if there are already *m* or more processes in the critical section. A process calls `leave()` to indicate it finished the critical section. If there is a process blocked waiting to enter, that process will be unblocked and allowed to continue, provided there are no more than *m* processes in the critical section.

Implement the `enter()` and `leave()` function. Use semaphores.

PhD Qualifying Examination

Fall 2005

Operating Systems

Question I: (100 points)

1. (70 points) Compute the number of page faults that may occur in a demand-paging system, using the optimal page-replacement algorithm, given that there are four physical frames available with the following reference string. Show work.

   1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 3, 4, 5, 3, 4, 5, 5, 1, 2, 3

2. (30 points) Describe how semaphores can be used to coordinate the operation of accessing shared resources. For example, how can a semaphore be used to prevent two processes from updating a shared memory at the same time?

Ph.D. Qualifying Examination

Fall 2005

Operating Systems

Question II (100 points)

1. (50 Pts.)
Please answer the following questions:

a. (25 Pts.)
Please explain what the Process Control Block is and describe the information it contains.

b. (25 Pts.)
Explain what the O.S. does when it creates a new process with the fork/exec mechanism.

2. (50 Pts.)
Consider a CPU with a TLB that caches 512 page table entries. A mapped memory access (during a TLB hit) takes 1200 ps. One access to main memory takes 1000 ps. What are the TLB access time and the TLB hit rate such that the effective memory-access time is 1250 ps ?
(Show all computations for full credit.)