Operating systems midterm examination-Answers

1.(10 points) Explain how the separation of policy and mechanism aids in building microkernel based operating systems.

Separation of policy and mechanism allows OS designers to implement a

small number of basic primitives in the kernel. These primitives are sim-

pliﬁed, because they are not dependent of any specific policy. They can then

be usedlevel.

2. (15 points) Suppose that we have a message passing system using mailboxes. When sending to a full mailbox or trying to receive from an empty one, a process does not block. Instead, it gets an error code back. The process responds o the error code by just trying again, over and over, until it succeeds. Does this scheme lead to race conditions?

It does not lead to race conditions (nothing is ever lost), but it is effectively

busy waiting. The error message provides feedback which prevents a race from happening.

3. (30 points) Consider a system of n processes, each of which has a unique priority number. Write a monitor that allocates three identical line printers to these processes, using the priority numbers for deciding the order of allocation.

type resource=monitor

 var P: array[0..2] of boolean

 X:condition

 procedure acquire (id: integer, printer-id:integer)

 begin

 ii P[0] and P[1] and P[2] then X.wait(id)

 if not P[0] then printer.id=0

 else if not P[1] then printer.id=1

 else printer.id=2

 P[printer.id]=true

 end

 procedure release (printer.id:integer)

 begin

 P[printer.id]=false

 X.signal

 end

 begin

 P[0]=P[1]=P[2]=false

 end

4. (10 points) An interactive system using round-robin scheduling and swapping tries to give guaranteed response to trivial requests as follows: after completing a round-robin cycle among all active jobs, the system determines the quantum for the next cycle by dividing a maximum response time by the number of jobs requiring service. Is this a practical policy?

It works only if there aren’t too many jobs in the system. If the quantum is decreased to satisfy more users in the system, it might easily get so small that a job will require a large number of quanta to be completed. This defeats the purpose of the algorithm.

5. (10 points) The shortest job next algorithm minimizes the average response time. Prove this for a batch of jobs which arrive at the same time with service times t1≤t2≤ … ≤ tn

ignoring further arrivals.

N users must wait for the execution of job 1, n-1 users must wait for the execution of job 2, and so on. Hence the average response time is

 (n\*t1 + (n-1)\*t2 +……+tn)/n

If we make any changes to this schedule, for example by exchanging jobs j and k (j<k) then the average response time is increased by (k-j)\*(tk-tj)/n ≥ 0

This means that the response time can only increase if the shortest time next algorithm in not used.

6. (15 points) Suppose a paging system has 2g+h  virtual addresses and 2 h+k  locations in primary memory for integers g,h and k. Write an expression for the implied page size of the system. How many bits are required to store a virtual address?

Since the virtual address size is the page size times the number of pages, then it is reasonable to assume that 2g+h represents the fact that there are 2g pages or address per page, and 2h addresses per page or pages. Similarly, since the process uses 2 h+k addresses as a set of page frames of the same size, the number of page frames or the page frame size is 2h and the page frame size or the number of page frames allocated to the process is 2k. Since 2h appears in both expressions, and since a page frame must be the same size as a page, we infer that 2h is the page and page frame size, 2g is the number of pages in the virtual address space, and that 2k is the number of page frames allocated to the process.

 A virtual address is g+h bits wide.

7. (10 points) A computer whose processes have 1024 pages in their address spaces keeps its page tables in memory. The overhead required for reading a word from the page table is 5 nsec. To reduce this overhead, the computer has a TLB which holds 32 (virtual page, physical page frame) pairs, and can do a look-up in 1 nsec. What hit rate is needed to reduce the mean overhead to 2 nsec?

The effective instruction time is 1*h* + 5(1 − *h* ), where *h* is the hit rate. If we

equate this formula with 2 and solve for *h*, we ﬁnd that *h* must be at least

0.75.