UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09047 OPERATING SYSTEMS

Thursday 10 th May 2018

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO of the three questions. If more than two questions are answered, only QUESTION 1 and QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: C. Stirling
External Examiners: S.Rogers, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) State three advantages of loadable kernel modules. [*3 marks*]

   (b) Briefly define the notion of process. Sketch a diagram of process states and transitions, giving one-sentence definitions of each state and transition. [*8 marks*]

   (c) State the criteria to be satisfied by solutions to the mutual exclusion problem. [*5 marks*]

   (d) Recall that a `TestAndSet` function operates on a word of memory at address `addr` as follows: it reads `*addr` (the contents of `addr`), sets `*addr` to 1, and returns the previous value, all in one atomic operation. Using C, show how to use `TestAndSet` to protect a section of critical code, `X`, shared between several processes. [*3 marks*]

   (e) Now consider a standard linked list of integers:

```
typedef struct {
  int data;  /* data stored in this node */
  node *next; /* pointer to the next node in the list */
} node;
typedef node *list;
```

   Let `mylist` be a global, shared, variable containing a list. We can insert a new data element `d` at the front of the list by `insert(& mylist,d)`, where `insert` is the following standard procedure:

```
void insert(list *list_ptr, int d) {
  node *newnode_ptr = malloc(sizeof(node));
  newnode_ptr->data = d;
  newnode_ptr->next = *list_ptr;
  *list_ptr = newnode_ptr;
}
```

   (f) Show, by considering two threads each calling insert on the same list, that this code has race conditions. [*3 marks*]

   (g) Using `TestAndSet` , modify the insert code so that it is thread-safe, [*3 marks*]

2. (a) Buddy allocation is used by the kernel to allocate memory for its own use. Suppose that 32 pages of memory are being allocated via the buddy algorithm, with a minimum block size of 1 page and no maximum block size. Assume that all memory is initially free. Describe the evolution of the memory allocation structure as the following sequence of events happens. When a request cannot be satisfied, say so, and assume the requesting process is killed.

   1. process P1 requests 10 pages of memory.
   2. process P2 requests 5 pages of memory.
   3. process P3 requests 1 page of memory.
   4. process P4 requests 7 pages of memory.
   5. process P1 terminates.
   6. process P5 requests 3 pages of memory.
   7. process P3 terminates.

   [*4 marks*]

   (b) Paging is used on modern systems to prevent fragmentation.

      i. Draw a labelled diagram explaining how a logical address is translated into a physical address with paging hardware. You should include the TLB and page table in your diagram. [*4 marks*]

      ii. In a 128-bit machine with 8KB pages, how large is the page table? [*2 marks*]

      iii. Explain one method to reduce this size [*2 marks*]

   (c) Virtual memory is a mechanism that supports larger logical memory than physical memory by using physical memory as a page cache to disk. Consider a system with only 4 frames i.e. only 4 pages can sit in memory.

   A process iteratively accesses six pages in the following order:
   1,2,3,4,5,6,5,6,5,6
   Initially the six pages are stored only on disk

      i. If the OS employs the MOST Recently Used (MRU) page replacement algorithm how many page faults will there be after 1 iteration and 10 iterations? [*2 marks*]

      ii. If the OS were able to use the clairvoyant Beladys algorithm, how many page faults will there now be after 1 iteration and 10 iterations? [*2 marks*]

      iii. What realistic replacement algorithm has the same behaviour, in this example, as Beladys? Describe a case where this replacement algorithm performs poorly. [*3 marks*]

[QUESTION CONTINUES ON THE NEXT PAGE]

(d) Briefly describe one possible requirement for an architecture to be virtualisable. [*2 marks*]

(e) Why is paravirtualisation a more efficient form of hardware virtualisation? [*2 marks*]

(f) In what environment would you choose to deploy paravirtualised virtual machines? [*2 marks*]

3. (a) List, with one sentence definitions, 3 criteria that might be used to evaluate process scheduling algorithms. [*3 marks*]

   (b) Explain briefly the round-robin, static priority, and feedback (dynamic priority queues) methods of scheduling. [*6 marks*]

   (c) Consider a single-processor system in which processes are pre-emptively scheduled with a quantum of 1 unit, and are assigned priorities on creation which do not change. Suppose that three processes are created with the following properties:

   - P1 starts at time 0, has priority 1 (low), and needs to execute for 5 quanta to complete its work.
   - P2 starts at time 3, has priority 3 (high), and needs 4 quanta to complete its work.
   - P3 starts at time 4, has priority 2 (medium), and needs 10 quanta to complete its work.

   All processes run without invoking any blocking procedures. Draw a diagram to show which processes are executing at which times, and note the finishing time of each process. [*4 marks*]

   (d) Now consider a system as in the previous part, but with the following additional behaviour:

   - After executing for 1 quantum, P1 requests exclusive access to a resource R.
   - Once it obtains R, it executes for 3 quanta before releasing it and continuing with its remaining quantum of execution.
   - After executing for 1 quantum, P3 requests exclusive access to R.
   - Having obtained it, it executes for 1 quantum before releasing it and continuing.
   - When a process is waiting for a resource that is not available, it is blocked.

   Draw a similar diagram, noting the finishing times. Why might this be considered a problem? [*6 marks*]

   (e) When a block of data is written from user space to a file on disk, different parts of the OS are involved. Outline a typical trajectory from user space, through the OS to disk for such an operation. [*6 marks*]