

This exam is **closed book**. You can have **one sheet** (i.e., **two pages**) of notes.  
Please be specific and answer every part of every question.

1. What are the undesirable effects of letting the hand of the Clock page replacement policy move (a) *too fast* and (b) *too slow*? (2×5 points) What has Berkeley UNIX done about both issues? (2×5 points)
  - (a) When the hand moves too fast, the policy invalidates too many page, which generates too many context switches when user processes access these pages. For that reason, BSD Unix limits the scanning rate of the hand of its clock policy to 300/scans per second.
  - (b) When the hand moves too slowly, pages that have not been recently accessed remain in main memory too long. To speed up the expulsion of these pages, BSD Unix has added a second hand that follows the first hand at a fixed angle. Now the first hand clears valid bits and the second hand expels the pages that have not been accessed by the process after their valid bit had been cleared by the first hand..
2. Give *one* example of:
  - (a) an external pager in the Mach operating system: (5 points)  
the file system\_\_\_\_\_
  - (b) a memory object managed by that external pager: (5 points)  
files\_\_\_\_\_
3. How does the Berkeley UNIX Fast File System organizes its *i-node table*? (5 points) What is the advantage of this organization? (5 points) (*Hint: This question is about the global organization of the i-node table and not about the contents of individual i-nodes.*)

The Fast File System subdivides each disk partition—or file system—into groups of contiguous cylinders. Rather than having a single i-node table for the whole disk partition, they distribute it among all cylinder groups so that most files will have their i-node and their data blocks in the same cylinder group, thus eliminating—or shortening—disk seeks
4. How many lines will be printed by the following program? ( 5 points)

```
main() { fork(); fork(); printf("Done!\n") }
```

\_\_\_\_\_ 4\_lines

5. Consider a virtual memory system with 64 bit addresses and 8 kilobyte pages. What would be the best page table organizations to support:

(a) *Partial subblocking* with a subblocking factor of 2: (5 points for a correct diagram)

VPN
PPN +bitmap
Next Pointer

Each page table entry would occupy \_\_\_\_\_  $3 \times 8 = 24$  \_\_\_\_\_ bytes. (5 points)

(b) *Complete subblocking* with a subblocking factor of 2: (5 points for a correct diagram)

VPN
PPN0
PPN1
Next Pointer

Each page table entry would occupy \_\_\_\_\_  $4 \times 8 = 32$  \_\_\_\_\_ bytes. (5 points)

6. In conventional shared memory systems, critical sections that consist of a single machine language instruction do not have to be surrounded by lock/unlock—or request/release—pairs. Why? (5 points) Is this still true with Munin? (5 points) Justify your answer by considering the case of a critical section reading the value of a single shared variable and storing it into a local variable. (10 points)

In a conventional shared memory system, there is no need to surround single instructions by lock/unlock pairs because single instructions are normally uninterruptible. In Munin, the pair request/release plays a double role:

- (a) It provides mutual exclusion
- (b) It ensures that the contents of the virtual shared memory are made consistent at the time the processor issues a request and that the new values are propagated when the process issues the matching release.

Omitting the request/release pair around a critical section consisting of a single instruction is thus dangerous as we would not have any guarantee that the instruction will access the most current version of the data in the distributed shared memory.

7. What characterizes a *self-tuning* cache replacement policy? (5 points) Which feature(s) of ARC make that policy self-tuning? (10 points) (*Hint: You have more than enough space to answer the question*)

A self-tuning cache replacement policy has no parameters that can be set by the system administrator and can affect the performance of the policy. ARC is self-tuning because it has no parameter than can be set or modified by the user.