

Review Questions

Instructions: You are not required to turn in answers to these review questions, but you should do them on your own after you read this section's papers to test your knowledge. Use Piazza or the TAs if you have questions or get stuck on any of these.

SPIN paper

- 1) What features of Modula-3 are essential for implementing the extensibility features of SPIN? Why?
- 2) Strand is an abstraction provided by SPIN for processor scheduling. Explain what this means. In particular, how will a library OS use the strand abstraction to achieve its desired functionality with respect to processor scheduling?
[Hint: Think of the data structures to be associated by the the library OS with a strand.]
- 3) How can SPIN ensure that a given library OS does not become a CPU hog, shutting out other library OSes?
[Hint: Think of how SPIN can control "macro" resource allocation at start up time for a library OS.]
- 4) Repeat Q3 for physical memory.

Exokernel paper

- 1) A library OS implements a paged virtual memory on top of Exokernel. An application running in this OS encounters a page fault.

Walk through the steps from the time the page fault occurs to the resumption of this application.

Make any reasonable assumptions to complete this exercise (stating the assumptions).
- 2) "The high level goals of SPIN and Exokernel are the same." Explain why this statement is true.
- 3) Map the mechanisms in Exokernel to corresponding mechanisms in SPIN.
- 4) Explain how the mechanisms in Exokernel help to meet the high level goals.
- 5) Repeat Q4 for SPIN.

Liedtke paper

- 1) Sketch how you will implement a device driver (say for a simple

device like a keyboard; if you are really adventurous do for a disk) using

the primitives provided by Liedtke's microkernel.

[Hint: Assume address space A0 magically exists that exposes the physical memory. Assume the device registers are memory mapped at some well known physical addresses in high memory.]

- 2) What is the difference between a "thread" as defined in Liedtke's paper and a Strand in SPIN?
- 3) Liedtke argues that a microkernel should fully exploit whatever the architecture gives to get a performance-conscious implementation of the microkernel. With this argument as the backdrop, explain how context switching overhead of a microkernel may be mitigated in modern architectures. Specifically, discuss the difference in approaches to solving this problem for the PowerPC and Intel Pentium architectures. Clearly explain the architectural difference between the two that warrant the difference in the microkernel implementation approaches.
- 4) Explain the notions of "independence" and "integrity" in the L3 microkernel.
- 5) All the subsystems in L3 live in user space. Every such subsystem has its own address space. The address space mechanism "grant" allows an address space to give a physical memory page frame to a peer subsystem. Can this compromise the "integrity" of a subsystem? Explain why or why not.
- 6) "Trust in the L3 microkernel is transitive." Explain what this sentence means.

Xen Paper

- 1) Imagine a Linux OS top of Xen (XenoLinux). A user launches an application "foo" on top of XenoLinux. Walk through the steps that happen before foo actually starts running.

- 2) foo starts executing; it executes a blocking system call

```
fd = fopen("bar");
```

- (a) Show all the interactions between XenoLinux and Xen for this call. Clearly indicate when foo resumes execution.

Upon resumption, foo executes another blocking system call

```
fwrite(fd, bufsize, buffer);
```

- (b) Show all the interactions between XenoLinux and Xen for this call. Clearly indicate when foo resumes execution.

Don't worry about the exact syntax of the above calls.

- 3) Construct and analyze a similar example for network transmission and reception by foo.
- 4) All processes in XenLinux occupy the virtual address space 0 through VMMAx, where VMMAx is some system defined limit of virtual memory per process.

XenLinux itself is a protection domain on top of Xen and contains all the processes that run on top of it.

Given this how does XenLinux provide protection of processes from one another?

[Hint: Work out the details of how the virtual address spaces of the processes are mapped by XenLinux via Xen.]

- 5) Give three important motivations for OS virtualization.

VMWare ESX

- 1) What is the difference between VMWare workstation and VMware ESX server? Discuss how OS calls from a user process running inside a VM are handled in the two cases.
- 2) Discuss the difference in memory virtualization in VMware and Xen.
- 3) What are the approaches to reclaiming memory in a virtualized over-subscribed server? Discuss the pros and cons of each approach.
- 4) Describe how ballooning may be used with Xen.
- 5) Explain how page sharing works in VMWare.
- 6) Explain the allocation policy of VMWare.

general OS structures

- (a) Microkernel based OS will always be less efficient compared to Monolithic kernel. True or False? Explain your answer in detail.
- (b) To what extent has the idea of configurable system services ({{it a la}} SPIN and Exokernel) influenced commercial operating systems? (Note: For the sake of this question, any flavor of Unix and Microsoft Windows fall under commercial OS).
You have to clearly explain how they have been or have not been influential, and back such statements with facts and reasons.