

COSC 6360: Operating Systems MATERIALS ON THE FIRST FALL 2009 QUIZ

You are only responsible for the materials discussed in class as they are *summarized* in the handouts and discussed in the PowerPoint presentations. I expect you to understand these summaries and to be able to comment around them.

Always ask yourself *why* a specific technique was used and *which* problem it tried to solve.

UNIX

D. M. Ritchie and K. Thompson, "The UNIX time-sharing system," *The Bell System Technical Journal*, 57, 6 (1978), pp. 1905-1929.

M. K. McKusick, W. N. Joy, S. J. Leffler and R. S. Fabry, "A Fast File System for UNIX," *ACM Transactions on Computer Systems*, 2:3 (1984), 181-197.

J. S. Quarterman, A. Silberschatz, and J. L. Peterson, "4.2 BSD and 4.3 BSD as examples of the UNIX system," *ACM Computing Surveys*, 17, 4 (1985), pp. 379-418.

The handout is self-contained but lacks diagrams: consult also the PowerPoint presentation. You might skip the section on sockets but should pay particular attention to the Fast File System.

Virtual Memory

O. Babaoglu and W. Joy, "Converting a swap-based system to do paging in an architecture lacking page-reference bits," *Proc. 8th ACM Symposium on Operating Systems Principles*, (1981), pp. 78-86.

Focus on:

- *How Berkeley UNIX simulates the page-referenced bit in software. (We did not discuss why Babaoglu and Joy rejected other page replacement policies.)*
- *What they did to limit the context switch overhead.*
- *Why they introduced the **vfork()** system call.*
- *How the page replacement policy evolved since the early eighties and why.*

Start with the handout and consult the paper when you don't understand a specific paragraph.

R. Rashid, A. Tevanian, M. Young, D. Golub, R. Baron, D. Black, W. Bolosky and J. Chew, "Machine-independent virtual memory management for paged uniprocessor and multiprocessor architectures," *IEEE Transactions on Computers*, C-37, 8 (1988), pp. 896-905.

The paper is very hard to read. Start with the summary and the slides. Focus on:

- *The objectives of the system and its support for mapped files.*
- *The notion of memory object and how it affected the design of the address map.*
- *The concept of inheritance.*
- *The way Mach implements the UNIX **fork()** including its use of copy on write.*
- *The Mach page replacement policy.*

Review problems

1. How many lines will the following program print?

```
main() {
    fork();
    printf("Hello!\n");
    fork();
    printf("Goodbye!\n");
} // main
```

2. A UNIX file system has 8 kilobyte pages and i-nodes with 15 block addresses. How many file bytes can be accessed (a) directly from the i-node, (b) with one level of indirection, (c) with two levels of indirections, and (d) with three levels of indirections?

Solutions 1. 4. 2. 4096 KB, 32768 MB, 1 GB - 32768 MB - 4096 KB and none. 3. 32768 and 32768 bytes.