

CSCI 8530: Computer Science
Advanced Operating Systems
Spring 2020
Homework 5

Due on April 23, 2020

Please keep answers short. Every sub-answer should be limited within 3 lines.

You may either type your answers or write them by hand and upload it to the Canvas. Note that a "good effort" on the homework includes that all answers are in your own words (short sentences). DO NOT SHARE YOUR ANSWERS WITH OTHER STUDENTS OR ATTEMPT TO OBTAIN ANSWERS FROM OTHERS.

Questions

1. Virtual memory systems always try to maintain a “slush fund” of unused – or at least available – page frames. What algorithm is used – and when it is used – to add page frames to this collection of page frames? Don’t identify a specific algorithm, but instead identify the general name associated with the class of algorithms used for this purpose.
2. Find a system that uses asynchronous I/O, and identify the mechanism by which a running program is notified when the operation completes. Please explain which approach, synchronous or asynchronous, makes it easier to program?
3. In class, two separate bindings are discussed: one is the binding from a device name (e.g., CONSOLE) to a descriptor (e.g., 0) and the other is the binding from a device descriptor to a specific hardware device. Please explain how Linux performs the two bindings.
4. Which part of a device driver deals directly with interrupt handling? What part of a device driver can block? In which process does an interrupt handler execute?
5. What is the structure of interrupt code for “*interrupt dispatching*” and “*interrupt controller*” on the Galileo?
6. We talk about the two halves of a device driver, even though these parts are not of equal size. But both parts are important. Briefly identify the major functions of each half of the device driver for the Xinu “tty” device. Which half contains the interrupt handler?
7. Returning from the execution of an *interrupt handler* occurs after the effect(s) of the interrupt have been dealt with. In particular, an interrupt could awaken one or more processes – that is, cause them to move from a blocked state to the ready state. (a) Illustrate one case where multiple processes may be awakened as the result of an interrupt. (b) Indicate how the interrupt handler prevents an awakened process from taking control of the processor before the interrupt handling has been completed.
8. Please briefly describe the hardware organization (e.g., registers) of a programmable interval timer. How is such a device usually employed by an operating system?

9. How do most operating systems avoid having to perform a $O(N)$ search for work to be done when the clock ticks? What is the computational complexity associated with the identification of tasks waiting for a particular time to arrive?
10. In Xinu sleep primitives, does *sleepms(3)* guarantee a minimum delay of 3 milliseconds, an exact delay of 3 milliseconds, or a maximum delay of 3 milliseconds?