

Examples of exercises

Use case: Photocopier driver

ARCOS

Operating Systems Design

Degree in Computer Engineering

University Carlos III of Madrid

Exercise

Statement (1/2)

- ▶ We start from a computer with one CPU. The aim is to develop a driver of a printer-scanner (with interruptions but without DMA) for a UNIX-like operating system with a monolithic non-preemptive kernel. The functionality is as follows:
 - ▶ The computer asks the printer-scanner for scanning a document, receiving the data in PDF format.
 - ▶ The computer asks the printer-scanner for printing a document, sending the data in PDF format.
 - ▶ The computer asks the printer-scanner for making photocopies (i.e. scanning a document and printing it without computer interaction).
 - ▶ The printer-scanner has only one interruption, indicating the end of one of these three operations.
 - ▶ Finally, the driver can be loaded and unloaded dynamically.

Exercise

Statement (2/2)

- a) Design the interface of the driver, including system calls.
- b) Define all data structures or modify the existing ones to provide the required functionality
- c) Implement using pseudo-code the required functionality. Which events are involved?

Exercise

solution

- a) The interface would be as follows:
 - ▶ Manage de HW device
 - ▶ **HW_interruption_scanner_printer();**
 - ▶ Driver interface using UNIX (optional)
 - ▶ **desc = open(printer_name, flags);**
 - ▶ **res = close(desc);**
 - ▶ **res = read(desc, buffer, size);**
 - Make the printer-scanner do a scan. Return the result in buffer.
 - ▶ **res = write(desc, buffer, size);**
 - Make the printer-scanner print the data in buffer.
 - ▶ **res = ioctl(desc, operation, params);**
 - If operation = PHOTOCOPY make the printer-scanner do photocopies
 - ▶ Loading and unloading the driver
 - ▶ **Load_driver_printer_scanner();**
 - ▶ **Unload_driver_printer_scanner();**

Exercise

solution

- b) Data structures are:
- ▶ Data structures used to print:
 - ▶ Queue of print requests (PRINT.request), and pointer to the current (PRINT.current).
 - ▶ Blocked processes list (PRINT.blocked_list).
 - ▶ Data structures used to scan:
 - ▶ Queue of print requests (SCAN.request), and pointer to the current (SCAN.current).
 - ▶ Blocked processes list (SCAN.blocked_list).
- c) The events involved are the following:
- ▶ Printer-scanner interruption
 - ▶ Write system call (to print)
 - ▶ Read system call (to scan)
 - ▶ ioctl system call (to make photocopies)
-
- ▶ The pseudo-code is the following
 - ▶ We assume that ioctl is a non-blocking system call

Exercise

solution

HW_interruption_scanner_printer()

- ▶ Get operation from I/O registers
- ▶ If operation == PRINT // Print ends
 - ▶ Proc= ExtractFirstProcess(PRINT.blocked_list);
 - ▶ Proc.state = ready;
 - ▶ Enqueue(ready_state_queue, proc);
- ▶ If non empty(PRINT.requests)
 - ▶ PRINT.current = dequeue(PRINT.requests)
 - ▶ Send PRINT order.
- ▶ If operation == SCAN // Scan ends
 - ▶ Copy data to buffer
 - ▶ If (is_not_last_data())
 - ▶ Return
 - ▶ Proc= ExtractFirstProcess(SCAN.blocked_list);
 - ▶ Proc.state = ready;
 - ▶ Enqueue(ready_state_queue, proc);
 - ▶ If non empty(SCAN.requests)
 - ▶ SCAN.current = dequeue(SCAN.requests)
 - ▶ Send SCAN order.

Exercise

solution

_kernel_write_system_call()

- ▶ new_request = new request ()
- ▶ Fill new_request

- ▶ If device is free
 - ▶ new_request = current;
 - ▶ Send PRINT order
- ▶ Else
 - ▶ enqueue(PRINT.requests,new_request)

- ▶ Enqueue(PRINT.blocked_list, current);
- ▶ Current.state = BLOCKED,
- ▶ Old_current = current;
- ▶ current= Scheduler(); // ExtractFirstProcess(ready_state_queue);
- ▶ Current.state = EXECUTION;
- ▶ Swap_context(old_current,current);// Activator (dispatcher)

Exercise

solution

_kernel_read_system_call()

- ▶ new_request = new request ()
- ▶ Fill new_request

- ▶ If device is free
 - ▶ new_request = current;
 - ▶ Send SCAN order
- ▶ Else
 - ▶ enqueue(SCAN.requests,new_request)

- ▶ Enqueue(SCAN.blocked_list, current);
- ▶ Current.state = BLOCKED,
- ▶ Old_current = current;
- ▶ current= Scheduler(); // ExtractFirstProcess(ready_state_queue);
- ▶ Current.state = EXECUTION;
- ▶ Swap_context(old_current,current); // Activator (dispatcher)
- ▶ Copy data to user space

Exercise

solution

`_kernel_ioctl_system_call()`

- ▶ If operation = PHOTOCOPY
 - ▶ Send PHOTOCOPY order

Exercise solution

1. Initial approach:

1. Draw a diagram of initial system state
2. Modify the diagram to incorporate the exercise requirements

2. Answer the proposed questions

3. Review the answers

Examples of exercises

Use case: photocopier driver

ARCOS

Operating Systems Design

Degree in Computer Engineering

University Carlos III of Madrid