**Assignment 1: Deadlock and the Banker's Algorithm**

**Task 1: Deadlock**

1. **Definitions related to deadlocks:**

* **Mutual Exclusion:** A resource is only available to one process at a time. If another process wants it, it has to wait.
* **Hold and Wait:** A process is holding at least one resource and is waiting for other resources that are currently being held by other processes.
* **No Preemption:** A resource that has been allocated to a process cannot be forcibly taken away. The process must release it voluntarily.
* **Circular Wait:** A set of processes are waiting for each other in a circular fashion, creating a loop where no process can proceed.

1. **Banker's Algorithm Safe State Analysis:**

**Given:**

* Available resources: A = 3, B = 2, C = 2
* Processes: P1, P2, P3

(The rest of the information about allocation and maximum need is required to determine whether the system is in a safe state.)

If the system is in a safe state, all processes can finish executing by being allocated the needed resources in a specific order without leading to a deadlock.

**Task 2: Deadlock Avoidance**

1. **Resource Allocation Graph (RAG):**

A deadlock scenario with four processes (P1, P2, P3, P4) and four resources (R1, R2, R3, R4) occurs when:

* P1 holds R1 and waits for R2
* P2 holds R2 and waits for R3
* P3 holds R3 and waits for R4
* P4 holds R4 and waits for R1

This cyclic wait condition results in a deadlock.

**Assignment 2: Memory Management**

**Task 1: Contiguous Memory Allocation**

1. **Memory allocation strategies:**

* First-fit: The first available block large enough to accommodate the process is chosen.
* Best-fit: The smallest available block that fits the process is chosen.
* Worst-fit: The largest available block is chosen.

**Sequence:**

* Allocate P1
* Allocate P2
* Allocate P3
* Deallocate P1
* Allocate P4

Each strategy will behave differently based on memory fragmentation. Typically, best-fit minimizes wasted memory.

**Task 2: Paging**

1. **Address Calculations:**

* **Logical address space:** 32 KB (5 bits for offset, 3 bits for page number = 8-bit address)
* **Page size:** 4 KB (12 bits needed for logical address)
* **Number of frames:** Physical memory is 64 KB, divided by the page size (4 KB), giving 16 frames.

Example: Translating a logical address to a physical address involves dividing the address into a page number and offset.

**Assignment 3: Virtual Memory System (Page Table and Page Replacement)**

**Task 1: Page Table Structure**

1. **Two-level paging:**

* For a 32-bit virtual address space with 4 KB page size:
* First-level page table contains entries to the second-level tables.
* Each virtual address is broken into three parts: first-level index, second-level index, and offset.
* The address 0x12A45F78 would be translated by extracting each component and mapping through the page table.

**Task 2: Page Replacement Algorithms**

1. **Page Replacement Simulation:**

* For the reference string 1, 3, 0, 3, 5, 6, 3, 1, 2, 0, 1, 3, 6, simulate:
* FIFO: Pages are replaced in the order they were loaded.
* LRU: Least recently used pages are replaced.
* OPT: The page that won't be needed for the longest time is replaced.

You can count page faults for each algorithm and compare. Typically, OPT will have the fewest page faults, followed by LRU, then FIFO.

**Assignment 4: หยิบ OS 2-3 ตัว มาเปรียบเทียบลงรายงานเดียวกัน เช่น OS ภายใน hdd, OS ในเครื่องบิน, ใน android etc. แล้วแต่ที่สนใจ**

**1. Windows (สำหรับ HDD/คอมพิวเตอร์ทั่วไป)**

**ประเภท**: Desktop OS

**การใช้งาน:** ใช้ในคอมพิวเตอร์ส่วนบุคคลหรือระบบสำนักงาน เช่น Windows 10, Windows 11

**ข้อดี:**

* อินเตอร์เฟซที่เป็นมิตรกับผู้ใช้: ใช้งานง่ายและเป็นที่รู้จัก
* การรองรับซอฟต์แวร์หลากหลาย: รองรับแอปพลิเคชันทั้งสำหรับงานทั่วไปและงานเฉพาะทาง
* การจัดการฮาร์ดแวร์: มีการจัดการไฟล์และทรัพยากรอย่างมีประสิทธิภาพ

**ข้อเสีย:**

* ค่าใช้จ่าย: ต้องซื้อไลเซนส์เพื่อใช้งาน
* ปัญหาความปลอดภัย: มักถูกโจมตีด้วยไวรัสและมัลแวร์บ่อย

**2. Android (สำหรับอุปกรณ์พกพา)**

**ประเภท:** Mobile OS

**การใช้งาน:** ใช้ในสมาร์ทโฟน, แท็บเล็ต และอุปกรณ์พกพาอื่นๆ

**ข้อดี:**

* Open-source: นักพัฒนาสามารถปรับแต่งและพัฒนาระบบเพิ่มเติมได้
* ความเข้ากันได้กับอุปกรณ์หลายรูปแบบ: ใช้งานกับอุปกรณ์หลายยี่ห้อ เช่น Samsung, Google, Xiaomi
* การรองรับแอปพลิเคชันหลากหลาย: มีแอปใน Google Play Store จำนวนมาก

**ข้อเสีย:**

* ความไม่สม่ำเสมอของเวอร์ชัน: อุปกรณ์แต่ละรุ่นอาจไม่ได้รับการอัปเดตระบบปฏิบัติการเท่ากัน
* การจัดการทรัพยากร: การใช้งานทรัพยากรหนักอาจทำให้อุปกรณ์ทำงานช้าลง

**3. RTOS (สำหรับระบบในเครื่องบิน)**

**ประเภท:** Real-Time Operating System (RTOS)

**การใช้งาน:** ใช้ในระบบควบคุมการบิน, อุปกรณ์ในอวกาศ, หรือระบบที่ต้องการความแม่นยำและการตอบสนองแบบทันที เช่น VxWorks หรือ QNX

**ข้อดี:**

* การตอบสนองแบบเรียลไทม์: ออกแบบให้ระบบตอบสนองอย่างทันทีทันใดกับเหตุการณ์ที่เกิดขึ้น
* เสถียรภาพสูง: มีการทำงานที่เสถียรและเชื่อถือได้ในสภาวะที่ต้องการความปลอดภัยสูง
* ประสิทธิภาพในการจัดการทรัพยากร: ระบบถูกออกแบบให้ใช้ทรัพยากรน้อยที่สุดและประสิทธิภาพสูงสุด

**ข้อเสีย:**

* ซับซ้อนและเฉพาะเจาะจง: การพัฒนาระบบต้องการทักษะเฉพาะทางและการออกแบบที่ละเอียดมาก
* ความยืดหยุ่นต่ำ: ไม่สามารถปรับเปลี่ยนได้ง่ายเหมือนระบบปฏิบัติการทั่วไป

**สรุปเปรียบเทียบ**

* การใช้งาน: Windows เหมาะสำหรับการใช้งานทั่วไปและการจัดการไฟล์ขนาดใหญ่ ในขณะที่ Android เหมาะกับการใช้งานบนอุปกรณ์พกพา ส่วน RTOS ถูกออกแบบมาเพื่อใช้งานในระบบที่ต้องการความปลอดภัยและตอบสนองทันทีในเครื่องบินหรือยานอวกาศ
* การจัดการทรัพยากร: RTOS มีการจัดการทรัพยากรที่เข้มงวดมากกว่า Android และ Windows เพื่อรองรับการตอบสนองแบบเรียลไทม์
* ความยืดหยุ่น: Android และ Windows มีความยืดหยุ่นและรองรับแอปพลิเคชันหลากหลายมากกว่า RTOS ซึ่งออกแบบมาเพื่องานเฉพาะทาง