ANSWERS

Chapter 1: BACKGROUND OF OPERATING SYSTEMS

Exercises:

1. List and briefly define the four main elements of a computer. (Q 1.1)

   Answer:
   - ❑ A **main memory**, which stores both data and instructions
   - ❑ an **arithmetic and logic unit** (ALU) capable of operating on binary data;
   - ❑ a **control unit**, which interprets the instructions in memory and causes them to be executed;
   - ❑ **input and output (I/O) equipment** operated by the control unit.

2. Define the two main categories of processor registers. (Q 1.2)

   Answer:
   User-visible registers: Enable the machine- or assembly-language programmer to minimize main memory references by optimizing register use. For high-level languages, an optimizing compiler will attempt to make intelligent choices of which variables to assign to registers and which to main memory locations. Some high-level languages, such as C, allow the programmer to suggest to the compiler which variables should be held in registers.

   Control and status registers: Used by the processor to control the operation of the processor and by privileged, operating system routines to control the execution of programs.

3. What is an interrupt? (Q 1.4)

   Answer:
   An interrupt is a mechanism by which other modules (I/O, memory) may interrupt the normal sequencing of the processor.

4. What are the differences between a trap and an interrupt?
   Apakah perbezaan antara perangkap dan sampukan?

   - Interrupt – external to the execution of the current instruction, reaction to an asynchoronous externel event.

   - Trap – associated with the execution of the current instruction, handling of an error or an exception condition

5. What is cache memory? (Q 1.7)

   Answer:
   Cache memory is a memory that is smaller and faster than main memory and that is interposed between the processor and main memory. The cache acts as a buffer for recently used memory locations.

6. Explain the potential performance disadvantage of a microkernel operating system. (Q 4.13)

   Answer:
   It takes longer to build and send a message via the microkernel, and accept and decode the reply, than to make a single service call.

7. What is the basic form of communications between processes or threads in a microkernel operating system? (Q 4.15)

   Answer: Messages.

8. What is Operating System? Briefly explain three objectives of an operating system design.
   Apakah sistem pengoperasian? Terangkan secara ringkas tiga objektif reka bentuk sistem pengoperasian.

   Answers:

   Operating system is a program that controls the execution of application programs.

   - Convenience-Makes the computer more convenient to use
   - Efficiency - Allows computer system resources to be use in an efficient manner
   - Ability to evolve - Permit effective development, testing and introduction of new system functions without interfering with service

9. What is the kernel of an operating system? What are the general functions of the kernel?
   Apakah ia kernel bagi sebuah sistem pengoperasian? Apakah fungsi-fungsi am bagi kernel?

   Answers:

   A portion of the operating system that includes the most heavily used portions of software. Generally the kernel is maintained permanently in main memory. The kernel runs in a privileged mode and responds to calls from process and interrupts from devices.

10. Give the difference between uniprogramming and multiprogramming.
    Berikan perbezaan di antara pengaturcaraan tunggal dan multipengaturcaraan .

    Answers:

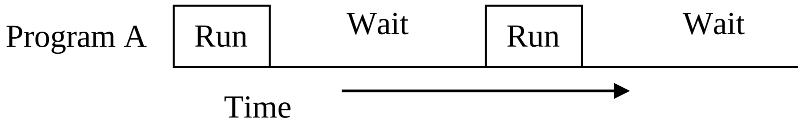
    Uniprogramming - Processor must wait for I/O instruction to complete before proceding

    Multiprogramming - When one job needs to wait for I/O, the processor can switch to the other job

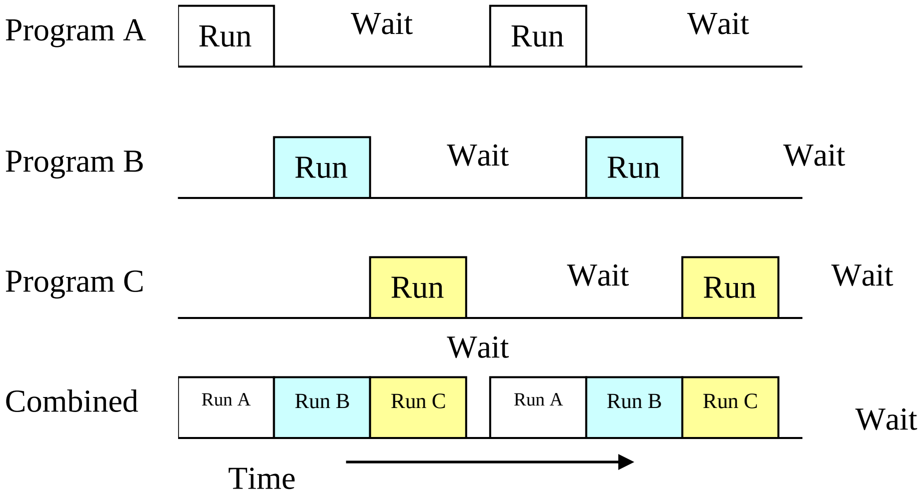
11. Draw some figures that show the operation examples of uniprogramming and multiprogramming with three programs.

Lukiskan beberapa gambar rajah yang menunjukkan operasi bagi contoh-contoh pengaturcaraan tunggal dan multipengaturcaraan dengan tiga atur cara.

Answers:



Uniprogramming



Multiprogramming with 3 programs

12. Define symmetric multiprocessing (SMP).
Takrifkan multi-proses simetrik.

Answers:

A symmetric multiprocessor can be defined as a standalone computer system with the following characteristics:

- There are multiple processor.
- These processors share the same main memory and I/O facilities, interconnected by a communication bus or other internal connection scheme.
- All the processor can perform the same functions (hence the term symmetric).

13. State THREE of the characteristics of modern operating systems.
Nyatakan ciri-ciri bagi sistem pengoperasian moden.

Answers: (Any 3 answers)

- Microkernel architecture,
- multithreading,
- symmetric multiprocessing,

- distributed operating systems and object-oriented design.

14. What is an operating system? Give two (2) examples of operating systems.
Apakah sistem pengoperasian? Berikan dua (2) contoh sistem pengoperasian.

Answers:

A program that controls the execution of application program.

Windows 2000, XP, ME,
Solaris, Unix Linux, Macintosh

15. What is multiprogramming and what is the main advantage?
Apakah multipengaturcaraan dan apakah kelebihan utamanya?

Answers:

Multiprogramming - When one job needs to wait for I/O, the processor can switch to the other job
fully utilization of processor

16. What are booting and bootstrap program?
Apakah maksud 'booting' dan atur cara 'bootstrap'?

Answer:
Booting – starting a computer by loading the kernel.

bootstrap program – a code stored in ROM that is able to locate the kernel, load it into memory and starts its execution.

- loaded at power-up or reboot
- Typically stored in ROM or EEPROM, generally known as **firmware**
- Initializes all aspects of system
- Loads operating system kernel and starts execution

17. What is the distinction between blocking and nonblocking with respect to messages?
Apakah perbezaan antara menunggu dan tidak-menunggu bila berkaitan dengan mesej?

Answers:

There are two aspects, the send and receive primitives. When a *send* primitive is executed in a process, there are two possibilities: either the sending process is blocked until the message is received, or it is not.

Similarly, when a process issues a *receive* primitive, there are two possibilities: If a message has previously been sent, the message is received and execution continues. If there is no waiting message, then either (a) the process is blocked until a message arrives, or (b) the process continues to execute, abandoning the attempt to receive.

18. Briefly explain the benefits of a microkernel organization.
Secara ringkas terangkan faedah organisasi 'microkernel'.

Answers:

1. Uniform interface on request made by a process
   - Don't distinguish between kernel-level and user-level services
   - All services are provided by means of message passing
2. Extensibility
   - Allows the addition of new services
3. Flexibility
   - New features added
   - Existing features can be subtracted
4. Portability
   - Changes needed to port the system to a new processor is changed in the microkernel - not in the other services
5. Reliability
   - Modular design
   - Small microkernel can be rigorously tested
6. Distributed system support
   - Message are sent without knowing what the target machine is
7. Object-oriented operating system
   - Components are objects with clearly defined interfaces that can be interconnected to form software

19. What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

An interrupt is a hardware-generated change of flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction;.
A trap is a software generated interrupt.
An interrupt can be used to signal the completion of an I/O to obviate the need for device polling.
A trap can be used to call operating systems routines or to catch arithmetic errors.

20. Direct memory access (DMA) is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.
   a. How does the CPU interface with the device to coordinate the transfer?
   b. How does the CPU know when the memory operations are complete?
   c. The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

The CPU can initiate a DMA operation by writing values into special registers that can be independently accessed by the device. The device initiates the corresponding operation once it receives a command from the CPU. When the device is finished with its operation, it interrupts the CPU to indicate the completion of the operation.

Both the device and the CPU can be accessing memory simultaneously. The memory controller provides access to the memory bus in a fair manner to these two entities. A CPU might therefore be unable to issue memory operations at peak speeds since it has to compete with the device in order to obtain access to the memory bus.

21. Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

Answer:

Caches are useful when two or more components need to exchange data, and the components perform transfers at differing speeds.

Caches solve the transfer problem by providing a buffer of intermediate speed between components. If the fast device finds the data it needs in the cache, it need not wait for the slower device.

The data in the cache must be kept consistent with the data in the components. If a component has a data value change, and the datum is also in the cache, the cache must also be updated. This is especially a problem on multiprocessor systems where more than one process may be accessing a datum.

A component may be eliminated by an equal-sized cache, but only if:
(a) the cache and the component have equivalent-saving capacity (that is if the component retains its data when electricity is removed, the cache must retain data as well), and
(b) the cache is affordable, because faster storage tends to be more expensive.

22. Define the essential properties of the following types of operating systems:

   a. Batch
   b. Interactive
   c. Time sharing
   d. Real time
   e. Network
   f. Parallel
   g. Distributed
   h. Clustered
   i. Handheld

Answer:

Batch – Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, off-line operation, spooling, and multiprogramming. Batch is good for executing large jobs that need little interaction; they can be submitted and picked up later.

Interactive – This system is composed of many short transactions where the results of the next transaction may be unpredictable. Response time needs to be short (seconds) since the user submits and waits for the result.

Time sharing – This system uses CPU scheduling and multiprogramming to provide economical interactive use of a system. The CPU switches rapidly from one user to another. Instead of having a job defined by a spooled card images, each program reads its next control card from the terminal, and output is normally printed immediately to the screen.

Real time – Often used in a dedicated application, this system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.

Network – Provides operating system features across a network such as file sharing.

SMP – Used in systems where there are multiple CPUs each running the same copy of the OS. Communication takes place across the system bus.

Distributed – This system distributes computation among several physical processors. The processors do not share memory or a clock. Instead, each processor has its own local memory. They communicate with each other through various communication lines, such as a high-speed bus or local area network.

Clustered – A clustered system combines multiple computers into a single system to perform computational tasks distributed across the cluster.

Handheld – A small computer system that performs simple tasks such as calendars, email, and web browsing. Handheld systems differ from traditional desktop systems with smaller memory and display screens and slower processors.

23. List five services provided by an OS that are designed to make it more convenient for users to use the computer system. In what case it would be impossible for user-level programs to provide these services? Explain.

- **Program execution**. The OS loads the contents (or sections) of a file into memory and begins its execution. A user-level program could not be trusted to properly allocate CPU time.
- **I/O operations**. Disks, tapes, serial lines, and other devices must be communicated with at a very low level. The user need only specify the device and the operation to perform on it, while the system converts that request into device- or controller-specific commands. User-level programs cannot be trusted to access only devices they should have access to and to access them only when they are otherwise unused.
- **File-system manipulation**. There are many details in file creation, deletion, allocation, and naming that users should not have to perform. Blocks of disk space are used by files and must be tracked. Deleting a file requires removing the name file information and freeing the allocation blocks. Protections must also be checked to assure proper file access. User programs could neither ensure adherence to protection methods nor be trusted to allocate only free blocks and deallocate blocks on file deletion.
- **Communications**. Message passing between systems requires that messages be turned into packets of information, sent to the network controller, transmitted across communications medium, and reassembled by the destination system. Packet ordering and data correction must take place. Again, user programs might not coordinate access to the network device, or they might receive packets destined for other processes.
- **Error detection**. Error detection occurs at both the hardware and software levels. At the hardware level, all data transfers must be inspected to ensure that data have not been corrupted in transit. All data on media must be checked to be sure they have not changed since they were written to the media. At the software level, media must be checked for data

consistency; for instance, do the number of allocated and unallocated blocks of storage match the total number of blocks on the device. There, errors are frequently process-independent (for instance, the corruption of data on a disk), so there must be a global program (the OS) that handles all types of errors. Also, by having errors processed by the OS, processes need not contain code to catch and correct all the errors possible on a system.