What two jobs do operating systems do in general?

1

What is a process?

2

What is the address space?

3

What is a thread?

4

What is multi-threading?

5

How do we make programs think they have sole use of memory?

6

What are the two most common modes of operation?

7

What are the three different approaches to engineering an OS?

8

A program in execution, the thread + address space.

2

1. *Manage the resources of the system.*
2. *Abstract the implementation of the system from the running programs.*

1

A sequence of instructions that are obeyed.

4

All memory locations the process can use.

3

*Use **relocation**, where we provide a virtual machine for each program, allowing them to behave as though they have sole use of memory. We need to support some kind of virtual memory for this.*

6

This is where we have multiple threads within the same process

5

Monolithic, layered and micro-kernels.

8

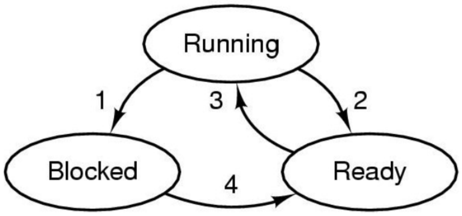
User and system

7

What are the three process states?

9

In the diagram, what is happening at each stage?



10

What is a PCB table?

11

In scheduling, what do the following mean?

1. *CPU burst*
2. *I/O burst*
3. *CPU bound*
4. *I/O bound*

12

What is a processes turnaround time?

13

What is a processes waiting time?

14

Briefly explain the first come first served scheduling algorithm.

15

Briefly explain the shortest remaining time first scheduling algorithm.

16

1. *Process needs to wait for I/O or event.*
2. *Process forcibly preempted -* ***interrupt / relinquish CPU / time-slice expired****.*
3. *Scheduler selects process to run.*
4. *I/O or event occurs.*

10

Running, ready, blocked

9

1. *Process executing on CPU*
2. *Process blocked, waiting for I/O*
3. *Long CPU bursts*
4. *Short CPU bursts*

12

Process control block, it contains all of the information needed about processes.

11

The time that the process waits to run.

14

The time from a process being submitted to it getting completed.

13

For each newly ready process, if CPU-burst is less than the time to complete the running process then context-switch and run the new process.

16

The first process in the ready state gets CPU time first. Once it is blocked or complete, the next process in the queue is run. Processes that require CPU time are added to the back of the queue.

15

What is meant by non-pre-emptive scheduling?

17

What is meant by pre-emptive scheduling?

18

What is the maximum length a process will be allowed to execute for called in pre-emptive processing?

19

What is process starvation?

20

In scheduling, what are static priorities?

21

In scheduling, what are dynamic priorities?

22

What is a race condition?

23

What do the following terms mean?

1. *Data inconsistency*
2. *Synchronisation*
3. *Critical section*
4. *Mutual exclusion*

24

A pre-emptive scheduler will temporarily interrupt a process, without requiring its cooperation, so that other processes can execute, and with the intention of resuming the task at a later time. The interrupted process is set in the 'ready' state so that it can be started again later.

18

Scheduling where processes run until they are terminated or blocked.

17

When the scheduling algorithm leaves a process out for a long time, causing the process to not receive any CPU time.

20

The 'time-slice' or 'time-quantum'.

19

Priorities that are assigned by the system to achieve certain goals.

22

Priorities that are predetermined for each process.

21

1. *Disagreement about data values*
2. *Using appropriate policies and mechanisms to ensure the correct operation of cooperating processes*
3. *Section of code in which shared data is used*
4. *At most 1 process can be in its critical section at once*

24

When one or more processes/threads execute in parallel, but the outcome depends on which finishes first.

23

What is deadlock?

25

What is the base register of a program?

26

What is the limit register of a program?

27

What is the base register usage sequence?

28

What is the limit register usage sequence?

29

What is the virtual address?

30

What performs the virtual to physical address conversion?

31

What are the two main reasons for virtual memory in a computer system?

32

A register that is loaded with the physical address where the program begins in memory.

26

Where there are a set of waiting processes where each process is waiting for something that can only be provided by another of the processes.

25

When the processor references memory, either fetch an instruction or read or write a data word, the CPU hardware automatically adds the base value to the address generated by the processor before sending the address out on the memory bus.

28

A register that is loaded with the length of the program.

27

An address that is generated by a program. It is converted to the actual 'physical address' which is used in memory.

30

When the base register usage sequence happens, the OS checks if the address offered is greater than the value in the limits register, in which case a fault is generated and access aborted.

29

1. *To allow a processor to address a much larger address space than is implemented by the physical memory*
2. *To support the OS in the management of processes*

32

The memory management unit (MMU)

31

What is a page table? 33

What are the two techniques used to implement virtual memory? 34

Segmented memory is split into chunks of a ______ size. 35

What is the formula to work out the number of pages used in a system? 36

What is the format of an address to paged memory? 37

What steps does the MMU take when it receives a paged address? 38

What columns are present in the page table? 39

In a page table, the resident flag checks whether the page is ______ 40

Paged virtual memory and segmented virtual memory.

34

A table used by the MMU to translate from a virtual to a physical address.

33

$$\#pages = \frac{addressSpace}{PageSize}$$

Where $addressSpace$ and $pageSize$ are measured in bits.

36

Segmented memory is split into chunks of a variable size.

35

1. *Look up the page number in the page table and see if it's in physical memory.*
2. *If it is in memory, generate a physical address page-base address + offset, and request that from RAM.*
3. *If it's not, abort the memory access, OS will load the page into memory (this is a page fault).*

38

Page number + offset

37

In a page table, the resident flag checks whether the page is currently in memory

40

- *Resident flag*
- *Used flag*
- *Dirty flag*
- *Physical address*
- *Disk address*

39

In a page table the used flag check whether the page

41

In a page table, what does the dirty flag do?

42

LRU is a page replacement algorithm, what does LRU stand for?

43

FIFO is a page replacement algorithm, what does FIFO stand for?

44

Briefly explain what the FIFO page replacement algorithm does.

45

Briefly explain what the second chance page replacement algorithm does.

46

Briefly explain the last recently used page replacement algorithm.

47

What is a write-back?

48

Checks whether the page has been written to while it's been in physical memory (so it needs to be copied back in full when it's put back onto the disk).

42

In a page table the used flag check whether the page has been used

41

First in first out.

44

Last recently used

43

This picks the oldest page with the fewest number of accesses since the last pass of the algorithm.

46

Identifies the oldest page in memory and gets rid of it.

45

When we write a page back to the disk when we swap it out.

48

This picks the page that has been used the longest ago. It's implemented using a timestamp that is updated when the page is read or written to.

47

What are two attributes that segments have that define how they are used?

49

When does a segment fault occur?

50

Explain external fragmentation.

51

What can we do to prevent fragmentation?

52

When the OS tries to access a segment that is not in memory.

50

Usage rights and access rights.

49

Use algorithms to find a good place to put segments, such as 'best fit' and 'first fit' when the OS is placing segments in memory.

52

When segments are moved in and out of main memory, 'holes' appear in the memory (due to segments having different sizes), which reduces the amount of useful memory.

51