Contents

[Threads 2](#_Toc86016329)

[2018 2](#_Toc86016330)

[2017 2](#_Toc86016331)

[2016 2](#_Toc86016332)

[2015 2](#_Toc86016333)

[2014 2](#_Toc86016334)

[Mutual Exclusion & Java Monitor & Threaded Servers 3](#_Toc86016335)

[2018 3](#_Toc86016336)

[2017 3](#_Toc86016337)

[2016 3](#_Toc86016338)

[2015 3](#_Toc86016339)

[2014 3](#_Toc86016340)

[Process Management 4](#_Toc86016341)

[2018 4](#_Toc86016342)

[2017 4](#_Toc86016343)

[2016 4](#_Toc86016344)

[2015 5](#_Toc86016345)

[2014 6](#_Toc86016346)

# Threads

### 2018

#### Explain the difference between the thread methods Wait, Sleep, and Yield in Java. Your answer should include coding examples on how each can be used.

**Wait**

This causes current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object. In other words, this method behaves exactly as if it simply performs the call wait(0).

**Sleep**

Thiscauses the currently executing thread to pause for a specified number of milliseconds. When one thread sleeps for a second, the other thread is scheduled.

**Yield**

This causes the thread to temporarily pause and allow other threads to run.

#### Compare and contrast the two ways of creating and starting threads in Java. Give example code to demonstrate your answer. Explain why a programmer may choose one way of creating threads over the other.

There are two ways to create a new thread of execution. One is to declare a class to be a subclass of Thread.

public class MyThread extends Thread {  
 public void run() {  
 System.*out*.println("Thread started running..");  
 }  
 public static void main(String args[]) {  
 MyThread mt = new MyThread();  
 mt.start();  
 }  
}

The other way to create a thread is to declare a class that implements the Runnable interface.

public void run() {  
 System.*out*.println("Thread started running..");  
}  
public static void main(String args[]) {  
 MyThread mt = new MyThread();  
 Thread t = new Thread(mt);  
 t.start();  
}

### 2017

#### Briefly explain the difference between a process and a thread.

Threads are like mini processes. The main difference between process and a thread is that threads normally share data (variables) while processes normally don’t. When several Java threads execute, the JVM can decide when and for how long to run each thread.

#### Explain what the thread methods start(), run(), yield(), sleep() and join() do.

* start() make the thread runnable
* run() implement this method
* sleep(long time) block the thread
* yield() let another thread run
* join() wait for a thread to finish

#### How can one obtain a thread-safe instance of a class such as HashMap (which is not itself thread-safe)? How should it be used correctly?

### 2016

#### Even the most modest of systems can benefit from considering performance and potential scalability in their design“. With reference to this statement explain the difference between performance and scalability.

Scalability is an essential component of enterprise software. Prioritizing it from the start leads to lower maintenance costs, better user experience, and higher agility. Software design is a balancing act where developers work to create the best product within a client's time and budget constraints.

#### State and explain Amdahl’s law. Using Amdahl’s law, calculate the max speedup with ten processors, for a program with 10% serialization. Would it be efficient to use 100 processors for this program? Explain your answer.

What is Amdahl's Law explain the speedup equation?

In general terms, Amdahl's Law states that in parallelization, if P is the proportion of a system or program that can be made parallel, and 1-P is the proportion that remains serial, then the maximum speedup S(N) that can be achieved using N processors is: S(N) = 1 / ((1-P) + (P/N))

If F is the fraction of a calculation that is sequential, and (1-F) is the fraction that can be parallelized, then the maximum speed-up that can be achieved by using P processors is 1/(F+(1-F)/P).

if If 90% of a calculation can be parallelized (i.e. 10% is sequential (serial)) 10% of a calculation can be parallelized then the maximum speed-up on 1000 processors is 1/(0.1+(1-0.1)/1000) or 9.9 (i.e. throwing an absurd amount of hardware at the calculation results in a maximum theoretical (i.e. actual results will be worse) speed-up of 9.9 vs a single processor).

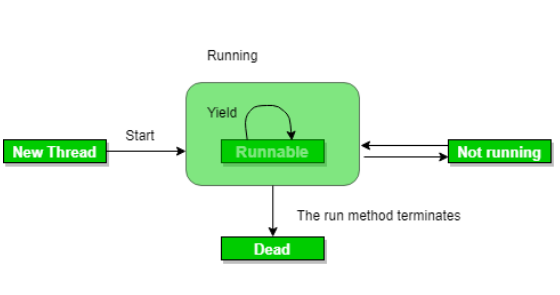
### 2015

#### Using short sections of java code, illustrate how to create and start a number of named Threads. The threads should loop 20 times printing out their name each time.

public class SimpleThread extends Thread  
{  
 public SimpleThread(String str) {  
 super(str);  
 }  
 public void run() {  
 for (int i = 0; i < 20; i++) {  
 System.*out*.println(i + 1 + " " + getName());  
 try {  
 *sleep*((1000));  
 }  
 catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 System.*out*.println("DONE! " + getName());  
 }  
}

public class SimpleThreadTest  
{  
 public static void main(String[] args) {  
 new SimpleThread("Dog").start();  
 new SimpleThread("Cat").start();  
 new SimpleThread("Fish").start();  
 }  
}

#### Draw a diagram showing all possible sequences of states through which a thread may pass. With reference to the thread you wrote in part (a) above explain why each state transition may occur



#### What is a Critical Section and what is mutual exclusion?

A piece of code in a process or thread that accesses a resource (memory for example) that is shared with another process or thread is called a critical section.

#### Give an example of hardware support provided by processors to enable the implementation of mutual exclusion. Explain why this support is so useful

It is very difficult to implement mutual exclusion without special purpose hardware support. Most (all) modern processors provide a special hardware instruction to support the implementation of mutual exclusion.

### 2014

#### Briefly describe the difference between a process and a thread.

The main difference between process and a thread is that threads normally share data (variables) while processes normally don’t.

#### Explain three reasons why developers use threads and concurrency.

These threads share the process's resources, but are able to execute independently. The threaded programming model provides developers with a useful abstraction of concurrent execution. Multithreading can also be applied to one process to enable parallel execution on a multiprocessing system.

# Mutual Exclusion & Java Monitor & Threaded Servers

### 2016

#### Briefly explain the terms Critical Section and Mutual Exclusion.

**Critical Section**

A piece of code in a process or thread that accesses a resource (memory for example) that is shared with another process or thread is called a critical section.

**Mutual Exclusion**

To ensure mutual exclusion it is necessary to ensure that when one process or thread is executing its critical section, no other process or thread can execute its critical section. A mutual exclusion is a program object that prevents simultaneous access to a shared resource.

#### The monitor is a key mechanism in Java for ensuring mutual exclusion. Explain how it works.

In concurrent programming (also known as parallel programming), a monitor is a synchronization construct that allows threads to have both mutual exclusion and the ability to wait (block) for a certain condition to become false.

An object whose methods can be safely used in a multithreaded environment. A class that has synchronized methods and uses the wait() and notify() methods like this is known as a Monitor. This is a concurrency mechanism that is directly implemented in Java.

# Process Management

### 2018

### 2017

#### In Process Management what is a context switch? Explain how an Operating System can implement a context switch. Your answer should include a description of the run and blocked queues.

Context switching is a process that involves switching of the CPU from one process or task to another. In this phenomenon, the execution of the process that is present in the running state is suspended by the kernel and another process that is present in the ready state is executed by the CPU.

#### Give two classifications of processes and explain how a Processing Scheduling algorithm should treat the two types of process.

#### Illustrate a Process State transition diagram. Explain the purpose of each state and possible transitions from state to state. Explain how the type of process e.g. MS Word\Media Player can effect state transitions.

Diagram

Description automatically generated

Word is started. First the program is loaded (copied) into memory. Its initial state is ready. Shortly afterwards the program is dispatched, and it starts executing (running). A splash screen is displayed on the screen and then the Word window itself appears. Then the program enters the blocked state. It is waiting for you to type. You type one character. That character is delivered to the program Word and its state is changed from blocked to ready (wakeup). At some stage the operating system dispatches the program (state changes to running) and the Word program decides and puts that character on the screen. Then the process state changes from running to blocked again (sleep). Suppose a Media Player was executing at the same time. While the Word program is blocked, the Media Player can execute. [If you have run a media player and a number of programs on a machine that is not fast enough, it can happen that the Media Player hasn’t time to decompress the media file in time and you can hear the results.]

### 2016

#### Define the following terms: Program, Process and Program Counter.

**Program**

In computing, a program is a specific set of ordered operations for a computer to perform.

**Process**

The simplest definition of a process is “a program in execution”

**Program Counter**

A program counter (PC) is a CPU register in the computer processor which has the address of the next instruction to be executed from memory. It is a digital counter needed for faster execution of tasks as well as for tracking the current execution point. A program counter is also known as an instruction counter, instruction pointer, instruction address register or sequence control register.

#### Briefly explain the difference between process scheduling and dispatching. Give two classifications of processes and explain how a Processing Scheduling algorithm should treat the two types of process.

Dispatching is a mechanism. The first process is taken off the ready queue and context switched into the CPU. Scheduling is a policy. It is concerned with ordering the ready queue so that there is optimum utilization of the CPU.

#### Briefly describe the difference between a process and a thread. Explain three reasons why developers use threads and concurrency.

1. Process means a program is in execution, whereas thread means a segment of a process.
2. A Process is not Lightweight, whereas Threads are Lightweight.
3. A Process takes more time to terminate, and the thread takes less time to terminate.
4. Process takes more time for creation, whereas Thread takes less time for creation.
5. Process likely takes more time for context switching whereas as Threads takes less time for context switching.
6. A Process is mostly isolated, whereas Threads share memory.
7. Process does not share data, and Threads share data with each other.

### 2015

#### Define the following terms: process, volatile environment, and context switching.

**Context Switching**

Context switching is a process that involves switching of the CPU from one process or task to another. In this phenomenon, the execution of the process that is present in the running state is suspended by the kernel and another process that is present in the ready state is executed by the CPU.

**Volatile Environment**

PCBs for each process, contain all the information required by the OS to manage processes in the system. It allows the operating system to restart a process. In order to do this the contents of all registers must be saved (e.g. LD R1, X). This is often known as the volatile environment of a process.

**Process**

The simplest definition of a process is “a program in execution”

#### Explain the Fetch-Execute cycle. Your answer should include explanations of the following terms: PC and CIR

1. Initialize

When a program is to be executed, the PC is set to contain the memory location of the first instruction of the program. A program counter (PC) is a CPU register in the computer processor which has the address of the next instruction to be executed from memory. It is a digital counter needed for faster execution of tasks as well as for tracking the current execution point.

1. Fetch

Get the machine instruction (contents of the memory location) pointed to by the PC and store it in the Current Instruction Register. Increment PC

1. Execute

Execute the instruction in the current instruction register (CIR).The current instruction register holds a machine instruction that is currently being executed

#### Briefly explain the difference between process scheduling and dispatching. Give two classifications of processes and explain how a Processing Scheduling algorithm should treat the two types of process.

Dispatching is a mechanism. The first process is taken off the ready queue and context switched into the CPU. Scheduling is a policy. It is concerned with ordering the ready queue so that there is optimum utilization of the CPU.

### 2014

#### Explain the Fetch-Execute cycle. Your answer should include explanations of the following terms: PC and CIR

1. Initialize

When a program is to be executed, the PC is set to contain the memory location of the first instruction of the program. A program counter (PC) is a CPU register in the computer processor which has the address of the next instruction to be executed from memory. It is a digital counter needed for faster execution of tasks as well as for tracking the current execution point.

1. Fetch

Get the machine instruction (contents of the memory location) pointed to by the PC and store it in the Current Instruction Register. Increment PC

1. Execute

Execute the instruction in the current instruction register (CIR).The current instruction register holds a machine instruction that is currently being executed

#### In Process Management what is a context switch? Explain how an Operating System can implement a context switch. Your answer should include a description of the run and blocked queues.

A context switching is a process that involves switching of the CPU from one process or task to another. In this phenomenon, the execution of the process that is present in the running state is suspended by the kernel and another process that is present in the ready state is executed by the CPU.

All processes in the system must be in one three states ready (or runnable) the process is waiting for the CPU running the process is using the CPU Blocked (or asleep) the process is waiting for some event to happen (such as

#### “It is advantageous to give I/O bound processes a higher priority than CPU bound processes, as they use the CPU for short periods of time”. Explain how Dynamic Priority Scheduling can achieve this.

This can be achieved by assigning p = 1/f where f is the fraction of the last time slice used.

A CPU bound process will use all its time quantum and will be assigned a priority = 1. An I/O bound process might use 1/2 of its time slice and get p = 2.

# Other

### 2018

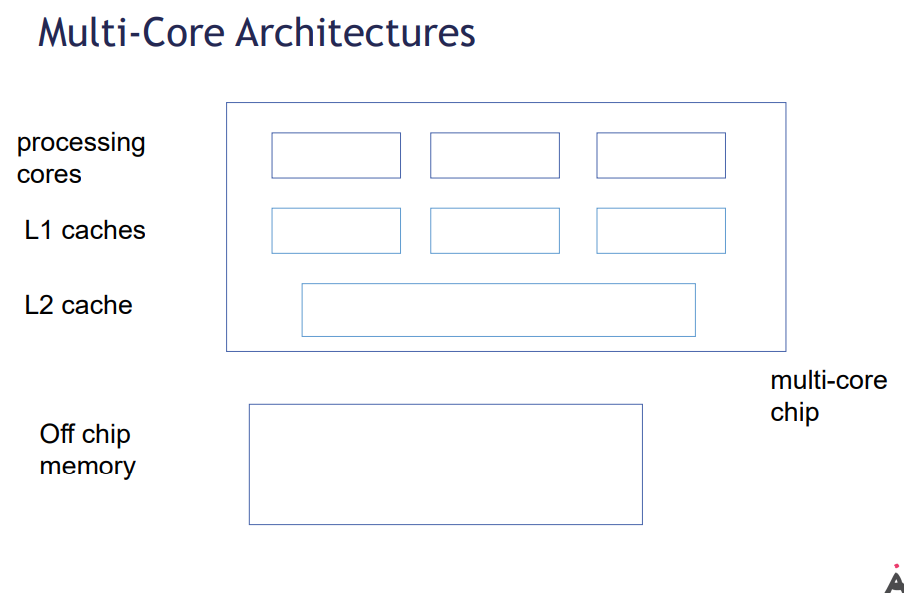
### 2017

#### Explain the following terms: main memory, cache memory, multi-core architecture. Use a diagram to illustrate your answer.

Main memory is a large array of words, indexed by address. Random Access Memory (CPU can go directly to any section of main memory and does not have go about the process in a sequential order.) • Depending on the platform, a word is typically either 32 or 64 bits, and so is an address. • A processor/core reads a value from memory by sending a message containing the desired address to memory.

Cache Memory • Closer to the processor or core and therefore faster. • When reading a memory location, first check the cache to see if it there. • L1 caches typically take 1 or 2 CPU cycles to access. • L2 caches typically take in the order of 10 cycles to access

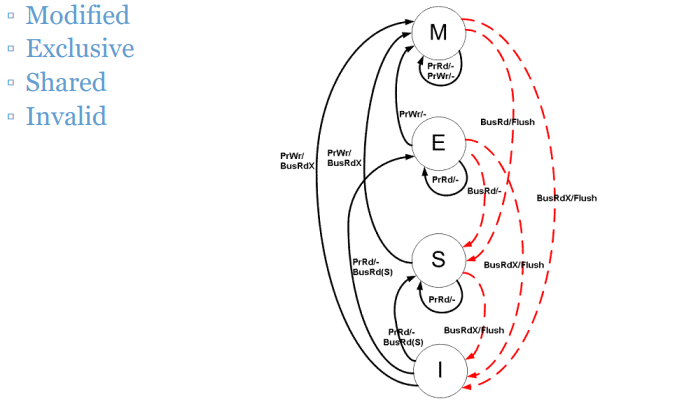
• Multicore architecture places multiple processor cores and bundles them as a single physical processor. • The proximity of multiple CPU cores allows the cache coherency circuitry to operate at a much higher clock rate than what is possible if the signals have to travel off-chip. • The ability of multi-core processors to increase application performance depends on the use of multiple threads within applications



#### Explain the function of a Cache Coherence algorithm.

A cache coherence algorithm ensures that data is not out of date. The MESI protocol works with 4 states of data in a cache

#### Describe the MESI protocol explaining the four states of data.



#### With regard to a concurrent application explain the term liveness. Briefly explain three common liveness issues for a concurrent application.

### 2016

#### Explain the Producer Consumer design pattern. Describe four benefits of this design pattern

#### Using short sections of java code illustrate

#### how a producer thread creates items and inserts them into an instance of java.util.concurrent.BlockingQueue.

#### how a consumer thread removes items from the BlockingQueue

#### a main() method that creates and starts a producer thread and a consumer thread.