

Operating System and concurrency, Lecture 3 Thread.

What is a thread? = It is a light-weight process, is a basic unit of CPU utilization. It shares code, data and ~~stack~~ stack and operating system resources.

Process vs thread = process vs thread

Isolated with its own virtual address.	shared virtual address space.
--	-------------------------------

Lot's of overhead	Less overhead.
-------------------	----------------

Benefits of multi-threading = Responsiveness

Resource sharing.

Multi-CPU utilization

Efficiency

2 ways of making thread. = extends Thread implements Runnable * run()

2 ways of create Thread. = 1) Thread P = new Thread();
1) extends Thread 2) implements Runnable

2) Thread q = new Thread(new PrintQ());

Do not call run() directly = Creating a new thread part is missed out and the run() method runs just as any other method.

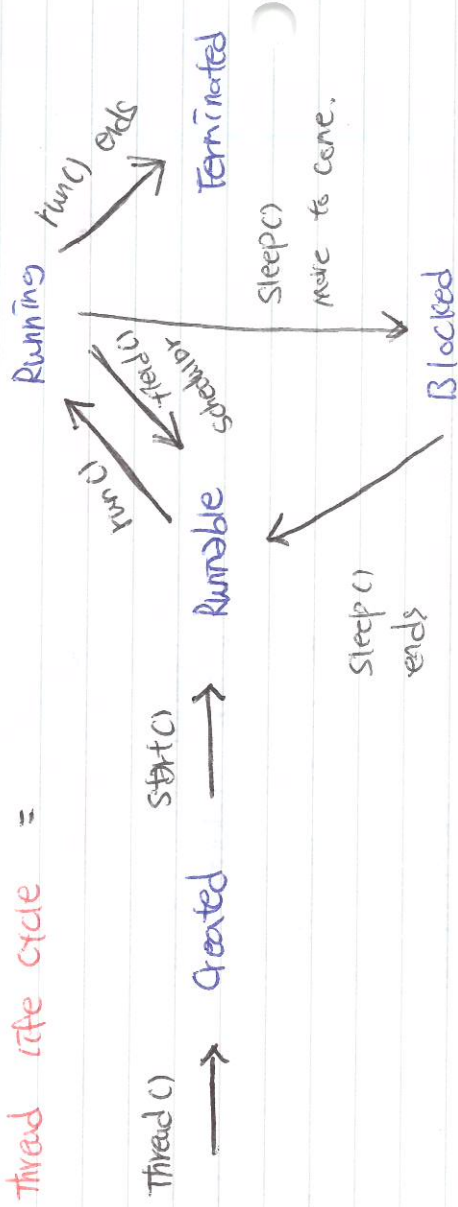
Interrupted Exception

- = Asynchronous cancellation
- thread is terminated immediately

Deferred cancellation

- The thread checks periodically to see whether it is to terminate.

Java thread life cycle



Race condition

= More than 2 processes compete to set shared resources.

Petersen's Algorithm
Semaphore

Atomicity

= An atomic statement is a single statement that cannot be interrupted

Critical Section

= In concurrent programming, concurrent access to shared resources can lead to unexpected behaviour so parts of the program where the shared resource is accessed is protected.

Synchronized

= Java provides the keyword `synchronized` which can be used to make a method as a critical section.

critical section

= signal that critical section is complete.

Synchronization Mechanism = preprotocol + postprotocol!

Solutions of critical section = Mutual Exclusion
 임의의 순간에 한 번만 들어갈 수 있다.

- only one process can enter the critical section at once.

Freedom from deadlock.

- If some processes are waiting to enter their critical section, eventually one must succeed.

Freedom from starvation

- If a process is waiting to enter its critical section it must do so eventually

Peterson's Algorithm.

= This Algorithm solves race condition but

• Busy waiting problem

• Starvation

// for 2 processes

Threads

임재우 - 소프트웨어 공학

2022. 9. 10.

Petersen's algorithm (2 processes) critical section

test & set

Baker's algorithm (N processes)



Mutual exclusion

Semaphore

Monitor

(cost performance issues)

(performance issues)