Concepts in Prog Langs – Ch. 14

/

What defines a concurrent program?

-

Two or more sequences of actions that may be executed simultaneously.

/

What are the two general ways a program may be executed concurrently?

-

1) Multiprogramming – interleaving on a single processor.

2) Multiprocessing – two or more processors may share memory or be connected by a network.

/

What are three terms to describe a sequence of actions?

What are their connotations?

-

1) Process – OS process with own address space

2) Thread – run under language run-time system, share same OS address space

3) Task – messages between actors

/

What is a main difficulty in designing concurrent systems?

-

Handling non-determinism: more than one possible sequence of actions for the same input.

/

What does every programming language with explicit concurrency provide?

-

Mechanisms to initiate and terminate independent sequential processes (tasks/thread).

/

What are three other mechanisms that concurrent programming languages provide some or all of?

-

1) communication between processes (channels, broadcasts, shared variables/objects)

2) coordination between processes (wait)

3) atomicity

/

What is message passing?

What are the main distinctions between forms of message passing?

-

A form of interprocess communication.

Buffering – buffered or unbuffered

Synchronicity – synchronous or asynchronous

Message Order – transmission order preserved or not

/

What is an atomic action?

-

An action that will either complete successfully or terminate in a state equal to the state prior to the initiation of the action.

/

What do you call a section of a program that accesses shared resources?

-

A critical section.

/

What is it called when only one process at a time is allowed to be in its critical section?

-

Mutual exclusion

/

What is busy waiting?

-

Using a loop to wait for a condition.

/

What is a deadlock?

-

When a process can never continue due to the state of another process.

/

How do processes avoid deadlock in the locking phase?

-

Agree on an ordering of lock acquisition and acquire all locks necessary to complete the transaction.

/

What are semaphores?

-

A concurrency mechanism without busy waiting. Integer flags are set, indicating the number of processes that may enter the critical, if at zero, the process is but in a waiting queue, until notification/signal that it may proceed.

/

What are monitors?

-

Synchronized objects with all synchronization details placed in the operations of the data type. The idea is to make sure that signaling happens and deadlock is avoided.

/

What are actors?

-

Reactive objects, with no shared state, that communicate through buffered asynchronous message passing. Each actor executes some computation in response to a message and sends out a reply when the computation is done. Message order is not guaranteed.

/

What are the three basic actions of an actor?

-

1) Send communications – to itself or other actors

2) Create actors

3) Specify replacement behavior (another actor)

/

What does one actor need to communicate with another actor?

-

A mail address.

/

What are the three main parts of a task?

-

1) A unique tag – distinguishing it from other tasks

2) A target – mail address of intended receiver

3) A communication – the data in the message

/

How does an actor change state?

-

By becoming another actor after processing input.