Lecture 7: Message Passing (concluded) and Linda

K. V. S. Prasad
Dept of Computer Science
Chalmers University

20 Sep 2011

Questions?

- Anything you want to say
  - Comments, questions, stray thoughts, etc.
  - Are we too fast/slow?
- Today’s ”and finally”
  - your views on this course and your education generally
- Practical problems?
  - Small extension for lab 1 deadline 1
    - Only to allow people to get into the system
  - Don’t miss deadlines! (you’re not Douglas Adams)

Plan for today

- Shared memory and message passing : recap
- Chap 8: Message passing
- Chap 9: Linda

Chap 3 & 4 (skipped for now)

REMINDER: exercises in Chaps. 1, 2, 3, 6, 7, 8

Comments on message passing

- Inter-process
  - Communication
  - Coordination
  - Cooperation
  - Competition
  - Concurrency
  - Synchronisation
- We mentioned simulation and examples such as pilots, athletes, dancers, musicians, …

Examples from the book

- Producer-consumer
  - Doesn’t matter whether synch/asynch
- Matrix-multiplication
  - Here, could be synchronous action : gangstepped
- Dining philosophers
  - With synchronous channels only.
  - Each fork behaves like a semaphore
  - Both deadlock and starvation seem possible!

Rendezvous

- Like synchronous channel, except
  - Addressing asymmetric
    - Sender knows receiver’s address (entry), not v-v.
  - The communication may involve computation and return of value by the receiver
  - So made for client-server

Ada

- Uses protected objects
  - Since the 1980's
    - though the concept was around earlier
  - Thus has the cleanest shared memory model
- Also has a very good communication model
  - Rendezvous
- Ada was decided carefully through the 1970s
  - Open debates and process of definition
- Has fallen away because of popularity of C, etc.
  - Use now seen as a proprietary secret!

Loosely coupled systems

- Tightly coupled systems
  - Shared memory
  - Synchronous communication
    - Whether one-to-one or broadcast
- Loosely coupled
  - Asynchronous
  - Persistent
- Linda is such a system
  - So are filing systems and databases?

Tuple space

- Large shared notice board
- Posted notes are in the form of tuples
- Can read notes matching any pattern
  - E.g., you look for a pair
    - Only singletons and triples posted
      - Block until someone posts a pair
- This blocking gives us synchronisation

Linda primitives

- Post(v1, v2, …, vn)
  - Put tuple of values out
  - Release an arbitrary proc waiting on this pattern
- Remove(x1, x2, …, xn)
  - X’s are variables
  - Remove an arbitrary matching note
  - Block if none available
- Read(x1, x2, …, xn)
  - Like remove, but leave note on board

Generalisation of read and remove

- Allow patterns such as (x, 4, y)
  - Matches only triples with middle element 4
- Allow patterns such as (x, c=, y)
  - Where c is a variable
  - Matches only triples with middle element = c

Linda examples

- From the book
- Given a monotonically increasing function f, with f(0) < 0 and f(1) > 0, find x where 0<x<1 such that f(x)=0.
  - Can be done by binary search
  - How to use more than one process
    - Can use ability to interrupt