Concurrency on Open Source Systems

Why do this course?

Concurrency has been a challenge for quite a few decades. Only recently has it emerged on the technological landscape. With the arrival of the Internet and multicore/multiCPU machines, there is a need to master programming these systems. Given the complexity of the challenge, this course will offer a mix of theoretical techniques and programming exercises. The Android system will be used as the vehicle to bring out concurrency issues. It offers an exciting departure from typical systems that will allow modeling some interesting issues in concurrency.

A concurrent programming system can be implemented on a resource-limited system by sharing resources between concurrent tasks. There are many ways of sharing resources and care must be taken to both provide the resources needed for each task when it executing and to avoid resource deadlocks between tasks. The Android architecture, popular now through its use on mobile telephones, provides specific ways of dealing with these problems.

This course will describe different kinds of concurrent programming problems through characteristic examples and show how they can be implemented on Android systems.

Course Prerequisites

First courses on: Operating Systems, Programming Languages

Course Outcome

- Understand concurrency in sequential and distributed computing
- Use concurrency thinking to design and modularise programs
- Understand and use modern IPC techniques for implementing concurrent systems

Course layout

The following table gives a rough layout of the topics to be covered in the course. Column 1 is a rough categorisation of the topics, and is useful only as a label in the time management given in the Gantt chart that follows. Column 2 is the approximate number of lecture hours, additively indicated, that would be needed to cover the topics. The topics themselves are listed in the third column.

/* Some comments are in C style in red. */

Unit#	No. lects	Lecture content
1	1	What is concurrency? /* Interactively bring out the ideas that students have */ Concurrency as a modularisation paradigm Challenges in concurrency; Illustrate using problems: Djikstra's "Deadly Embrace" description Dining philosophers/Cigarette Smokers/Sleeping Barber Some solutions using semaphores Prerequisites: Operating Systems, Programming Languages Android development setup instructions – Eclipse based See: developer.android.com
	+1	The Android Architecture overview (standard material off the web)
	+2	/* Questions to seed the rest of the course */ In how many different ways can programs be executed? In how many different ways can different program executions be expressed? In how many different ways can applications be structured? Can parts of a program be executed in parallel? Can a program be composed of parts of other programs?
	+3	Concurrency thinking at level of programmer intention: /* Motivate Maths */ Deadlocks and starvation, Synchronisation, Clocking problem in distributed systems, Global snapshot problem.
2	+5	Program expression strategies to express communication: Communication as data exchange. Notion of a channel & data to exchange Procedure call and its various activation techniques.
	+1	Syntax structure of the CCS/π calculus
	+3	Program organisation: Statements, expressions, procedures, classes, modules a. Procedure call b. Single caller – multiple callees c. Single callee – multiple callers d. Signals – event driven calls e. Coroutines – execution multiplexing by explicit switching f. Deep calls – exceptions – mainly "backward-down-the-activations" g. Network calls – RPC/RMI
	+1	Android approach to program expression: Java + XML, Multiple entry points
	+?	Implementation techniques of some: Optional – care needed to ensure that this component does not take us away from the course goals.
3	+3	Program execution models according to programmer intention: Sequential execution, co-routines, concurrent, and distributed. Context for program execution /* Process + OS = combinator? */
	+2	Program building: Tool chains and system software support for converting program expressions to realise program execution models above.
	+1	Program execution: The OS and its notion of a "process".

		/* OS = kernel + system software. Linux kernel + GNU system software = processes with single entry point, versus Linux kernel + Android system software = process with multiple entry points */
4	+1 +2 +2 +4	Communicating processes: Architectures – shared memory, messages Procedure calls – messages implementated via shared memory (stack) Client-server systems – message passing through “mail boxes” & RPC/RMI Client-server “Service discovery” – a message passing through Open binder
5	+6	Pseudo versus true multitasking: Sequential versus Distributed computing Graph theoretical modeling of resource deadlocks: the value of formal methods
	+2	Operational semantics of the CCS/π calculus
6	+4	Worked examples using a mathematical model like the π calculus, or CSP, ...
	Total: 44	

Lectures Gantt Chart:
1st row – Week number, 2nd row – Session number. 4th row – Section number (1st column of table)

1	2	3	4	5	6	7	8	9	10	11
	5	0	5	0	5	0	5	0		
	Concurrency: Thinking, Expression			Concurrency: Execution and methods						Maths
1		2		3		4		5		6

/* Needs rethinking. Will be reworked on after some practice on Android */
Assignments Gantt Chart:

Week ->

1	2	3	4	5	6	7	8	9	10
								π calculus probs	

Rationale:

1. Problem sheets for π calculus.

Suggested References:

1. **The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications**

Clay Breshears
ISBN-13: 978-0596521530 ISBN-10: 0596521537
http://www.amazon.com/The-Art-Concurrency-Parallel-Applications/dp/0596521537

2. **Programming Android**

G. Blake Meike
Rs. 577.00
http://www.amazon.in/Programming-Android-G-Blake-Meike/dp/9350239191?tag=googinhydr18418-21

3. **Professional Android 4 Application Development (Wrox)**

Reto Meier
Rs. 519.00
http://www.amazon.in/Professional-Android-Application-Development-Wrox/dp/812653608X/ref=pd_cp_b_0

4. **Concurrent Programming: Algorithms, Principles, and Foundations**

Michel Raynal
ISBN-13: 978-3642320262 ISBN-10: 3642320260 Edition: 2013th
$68.15
http://www.amazon.com/Concurrent-Programming-Algorithms-Principles-Foundations/dp/3642320260/ref=sr_1_22?s=books&ie=UTF8&qid=1413724046&sr=1-22&keywords=Concurrency

5. **Theory and Practice of Concurrency**

A. Roscoe
ISBN-13: 978-0136744092 ISBN-10: 0136744095 Edition: 1st
$24.23
http://www.amazon.com/Theory-Practice-Concurrency-Roscoe/dp/0136744095/ref=sr_1_23?s=books&ie=UTF8&qid=1413724046&sr=1-23&keywords=Concurrency

6. **Communicating and Mobile Systems: The π Calculus**

Robin Milner
ISBN-13: 978-0521658690 ISBN-10: 0521658691
$46.10
http://www.amazon.com/Communicating-Mobile-Systems-The-Calculus/dp/0521658691/ref=pd_sim_sbs_b_1?ie=UTF8&refRID=0ZSC376RWZ92VKDBMBPD