

ON SOFTWARE ISSUES  
IN  
DISTRIBUTED COMPUTING

BY

NIRAJ KUMAR SHARMA  
CENTRE FOR COMPUTER SCIENCE AND ENGINEERING

THESIS SUBMITTED  
IN FULFILMENT OF THE REQUIREMENTS OF  
THE DEGREE OF  
DOCTOR OF PHILOSOPHY

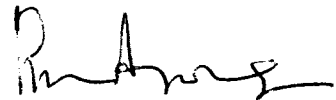
TO THE  
INDIAN INSTITUTE OF TECHNOLOGY, DELHI

JANUARY, 1983

CERTIFICATE

Certified that the thesis entitled 'ON SOFTWARE ISSUES IN DISTRIBUTED COMPUTING' being submitted by Mr. N.K. Sharma, is a record of work carried out by him under my supervision and guidance. The matter embodied in this thesis has not been submitted for the award of any other degree.

24 January, 1983



Prof. R.K. Arora

Computer Centre  
Indian Institute of Technology  
Delhi

## ACKNOWLEDGEMENTS

I am very grateful to my supervisor Professor R.K. Arora for his personal interest, constructive criticism, and constant encouragement.

My thanks are due to Professor M.K. Jain for his help and encouragement. My thanks are also due to the faculty and the other staff of the Computer Centre for their cooperation and help in many ways.

I would like to express my gratitude to Dr. S.P. Rana and all my friends who have helped me in several ways.



N.K. Sharma

**DEDICATED TO MY PARENTS**

## ABSTRACT

This thesis deals with some of the important software design issues related to distributed computing systems. The first part of the thesis describes two methodologies to design the distributed programming language which can be used to develop well-structured, reliable, and efficient distributed programs systematically. In one of the designs, the interprocess communication and synchronization are accomplished through procedure calls and in other one through input/output commands. A comparison of the suggested languages is carried out with other distributed programming languages available in the literature. Issues involved in the abstract implementation of these languages have also been discussed. A mechanism for writing terminating distributed programs has also been suggested which can be introduced in any distributed programming language.

In the second part of the thesis, a methodology to design the distributed operating system is discussed. The design of the distributed operating system has been expressed using both the distributed programming languages suggested in the thesis. It can tolerate processor or communication link failures and provide message delivery assurance.

## CONTENTS

	Page
ABSTRACT	i
CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 DISTRIBUTED COMPUTING SYSTEMS	4
1.2.1 CHARACTERSTICS	4
1.2.2 SOFTWARE DESIGN ISSUES	5
1.3 DISTRIBUTED PROGRAMMING LANGUAGES	6
1.3.1 FEATURES FOR DISTRIBUTED COMPUTATION	6
1.3.2 SURVEY OF RELATED WORK	7
1.4 DISTRIBUTED OPERATING SYSTEMS	9
1.4.1 PROPERTIES	9
1.4.2 DESIGN METHODOLOGIES	9
1.4.3 SURVEY OF A FEW EXISTING DISTRIBUTED OPERATING SYSTEMS	11
1.5 THESIS OVERVIEW	12
CHAPTER 2 PROCEDURE CALL BASED INTERPROCESS COMMUNICATION AND SYNCHRONIZATION	14
2.1 INTRODUCTION	14
2.2 LANGUAGE CONCEPTS	15
2.3 ILLUSTRATIVE EXAMPLES	21
2.3.1 INTEGER SEMAPHORE	21
2.3.2 BOUNDED BUFFER	22
2.3.3 DOUBLE BOUNDED BUFFER	22
2.3.4 SHORTEST JOB NEXT SCHEDULER	23

	2.3.5	DINING PHILOSOPHERS	24
	2.3.6	SORTING ARRAY	26
	2.3.7	CHARACTER STREAM	28
	2.3.8	READERS AND WRITERS	28
	2.3.9	VENDING MACHINE	29
	2.3.10	COROUTINES	30
	2.3.11	PRIME NUMBERS	31
	2.4	COMPARISON WITH OTHER LANGUAGES	32
	2.5	ABSTRACT IMPLEMENTATION	33
	2.6	CONCLUDING REMARKS	39
CHAPTER	3	INPUT/OUTPUT COMMAND BASED INTER- PROCESS COMMUNICATION AND SYNCHRONIZATION	40
	3.1	INTRODUCTION	40
	3.2	CRITICAL EVALUATION OF CSP	41
	3.3	LANGUAGE CONCEPTS	43
	3.4	ILLUSTRATIVE EXAMPLES	51
	3.4.1	INTEGER SEMAPHOR	52
	3.4.2	BOUNDED BUFFER	52
	3.4.3	DOUBLE BOUNDED BUFFER	53
	3.4.4	LIBRARY ROUTINE	54
	3.4.5	SHORTEST JOB NEXT SCHEDULER	55
	3.4.6	DINING PHILOSOPHERS	56
	3.4.7	SORTING ARRAY	58
	3.4.8	CHARACTER STREAM	60
	3.4.9	READERS AND WRITERS	60
	3.4.10	VENDING MACHINE	61

	3.4.11 COROUTINES	62
	3.4.12 PRIME NUMBERS	63
	3.5 COMPARISON WITH OTHER LANGUAGES	64
	3.6 ABSTRACT IMPLEMENTATION	65
	3.7 CONCLUDING REMARKS	70
CHAPTER	4 DISTRIBUTED TERMINATION	71
	4.1 INTRODUCTION	71
	4.2 SOLUTION OF DISTRIBUTED TERMINATION PROBLEM AND SUPPORTING LANGUAGE CONSTRUCTS	72
	4.3 CORRECTNESS OF THE SOLUTION	77
	4.4 ILLUSTRATIVE EXAMPLES	78
	4.4.1 BOUNDED BUFFER	78
	4.4.2 SORTING ARRAY	80
	4.4.3 DUTCH NATIONAL FLAG	82
	4.5 COMPARISON WITH OTHER PROPOSED SOLUTIONS	83
	4.6 CONCLUDING REMARKS	84
CHAPTER	5 DESIGN OF DISTRIBUTED OPERATING SYSTEM	85
	5.1 INTRODUCTION	85
	5.2 DISTRIBUTED OPERATING SYSTEM MODEL	86
	5.3 CONFIGURATOR	89
	5.3.1 FUNCTIONS AND STRUCTURE OF CONFIGURATOR	89
	5.3.2 A METHODOLOGY TO ESTIMATE IPC	90
	5.4 ASSIGNER	96
	5.4.1 DATA STRUCTURES	96
	5.4.2 THE STRUCTURE OF ASSIGNER	97

5.4.3	THE PROPERTIES OF ASSIGNER	98
5.4.4	IMPLEMENTATION STRATEGIES	99
5.4.4.1	CENTRALIZED CONTROL STRATEGIES	100
5.4.4.1.1	A CENTRALIZED ASSIGNER	100
5.4.4.2	DISTRIBUTED CONTROL STRATEGIES	106
5.4.4.2.1	MULTIPLE COPIES OF ASSIGNER	107
5.4.4.2.2	THE DISTRIBUTED PIPELINED ASSIGNER	109
5.5	EXECUTION CONTROLLER	113
5.5.1	ABSTRACT IMPLEMENTATION	114
5.5.1.1	DATA STRUCTURES	114
5.5.1.2	KERNEL PRIMITIVES	118
5.5.1.3	PROGRAM TEXT	119
5.6	CONCLUDING REMARKS	143
CHAPTER 6	CONCLUSIONS	144
6.1	SUMMARY	144
6.2	DIRECTIONS FOR FUTURE RESEARCH	145
	REFERENCES	146