# System Structure for Software Fault Tolerance

Author : Brian Randell

## Introduction

- Faults are common in most of the hardware and software systems
- Fault-tolerance will enable a system to continue operating properly in the event of failure of some of its components

## Fault Tolerant Techniques

- Recovery Blocks
- Conversations
- Fault-tolerant interfaces

## Fault tolerance structure

- Software fault tolerance structures are analogous to Stand-by sparing model of hardware
- What is Stand-by sparing model?

## Recovery Blocks

- Recovery block helps in error detection and recovery by providing extra information with their corresponding blocks
- Recovery block has two important characteristics :

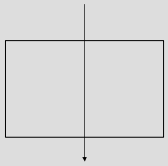
  1. Switching to the use of spare component
  2. Complexity of the system remains same

The recovery block consists of

1. Primary alternate
2. Acceptance Tests
3. Other alternates
4. Restoring the state of the system

## Primary Alternate

- It is the conventional program
- Enters the recovery block to perform the desired operation



Recovery block in single process

## Acceptance Tests

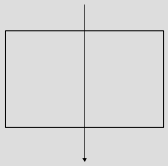
- Acceptance-Tests ensure the operation performed by the "alternates" is accepted
- Tests are performed with respect to the nonlocal variables

Example:

```
ensure sorted (S) and (sum(S)=sum(prior S)
 by quickersort(S)
  else by quicksort(S)
   else error
```

## Other Alternates

**Other** alternates enter the recovery block, if the primary alternate fails the acceptance test, to perform the operation



## Restoring the System state

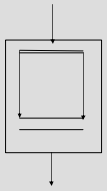
Recursive Cache Mechanism:

1. All the states of nonlocal variables with respect to the recovery block are saved in the recursive cache before the entry of alternate in to the block
2. Recursive cache is divided in to regions if nested recovery blocks are present

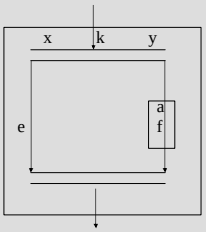
## Simple Recovery Block

```
A : ensure :AT
by                AP : begin
< pgm txt>
end
else by
AQ : begin
<pgm txt>
end                                  else error
```

## Recovery Block



Parallel process with in recovery block

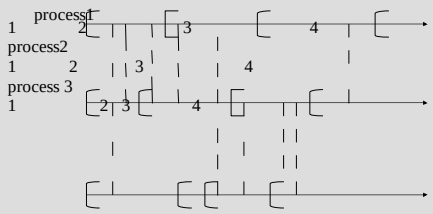



Parallel process with a nested recovery block

## Error Recovery - Interactive Processes

- If process interaction takes place, then the recovery block structure used for error recovery leads to “domino effect”.
- What is Domino Effect ?

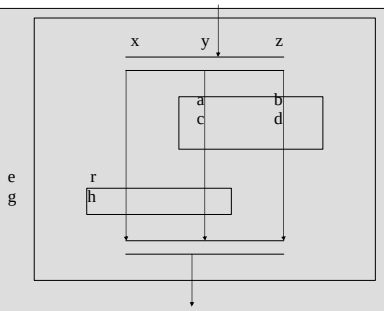
  1. Recovery blocks are uncoordinated
  2. Processes are symmetrical with respect to failure propagation




## Conversation

- Coordinates the recovery block structures of interacting processes
- A recovery block which span two or more processes is termed as “conversation”

## Parallel processes with conversations




## Multilevel Systems

- This is a method of structuring systems which uses asymmetrical failure propagation to avoid domino effect
- Consists of sequence of virtual machines
- Opaque Virtual interfaces

## Errors above a virtual machine Interface

- Every thing happens in a given level is the result of activity the level below is responsible
- Both progress and fall back in level i is the result of progress in level i-1

## Errors below a virtual machine Interface

- Deals with inverse operations
- Checking overall acceptability , in level 'i-1'

## Fault-Tolerant Virtual machine interfaces

- Complete Interpreter
- Conventional Interpreter
  Difference: Complete interpreter provides set of three related procedures rather than a single procedure

## Fault-Tolerant Virtual machine interfaces

The three procedures for Complete Interpreter are

1. Interpretation Procedure
2. Inverse Procedure
3. Acceptance Procedure

## Questions?