INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4.SMR. 405/15

SECOND WORKSHOP ON TELEMATICS

6 - 24 November 1989

Communication Software

J. GROMPONE
Interfase Ltd., Montevideo, Uruguay

COMMUNICATIONS SOFTWARE

COMMUNICATIONS EQUIPMENT:

- DEDICATED SYSTEM
- STORED PROGRAM CONTROL (SPC)
- PROGRAMMED INTEGRATED CIRCUIT
- INFORMATION PROCESSING SYSTEM

MAIN SUBJECTS:

- DESIGN
- MODELLING
- SPECIFICATION
- IMPLEMENTATION
- QUALITY CONTROL
- PROJECT DESIGN

SOFTWARE ENGINEERING

REAL TIME SYSTEMS

REAL TIME CONCEPTS

USUAL FORM:

- CLOCKS
- INTERRUPTS
- REAL-TIME CLOCKS
- REAL-TIME PROCESSES
- PARALELISM
- DISTRIBUTED PROCESSES

ABSTRACT FORM:

- CONCURRENCY
- EXCLUSION
- SYNCHRONIZATION

PROBLEM EVOLUTION

FIRST PROBLEMS 1955/60

- LARGE ASSEMBLER PROGRAMS

SOFTWARE CRISIS 1960/65

- /360 O.S.
- SPACE PROGRAMS

NEW IDEAS 1965/70

- DIJKSTRA
- SEMAPHORES
- CRITICAL REGION
- CRITICAL VARIABLES

NEW LANGUAGES 1970/85

- CONCURRENT STATEMENT
- SHARED VARIABLES
- CONCURRENT PASCAL (1974)
- MODULA (1977)
- CHILL (1980)
- ADA (1983)

CONCURRENCY

HARDWARE AMBIENT:

- ONE MACHINE + REAL-TIME CLOCK
- SEVERAL MACHINES

SOFTWARE AMBIENT:

GLOBAL VARIABLES

PROC. 1 PROC. 2 PROC. 3

EXECUTION:

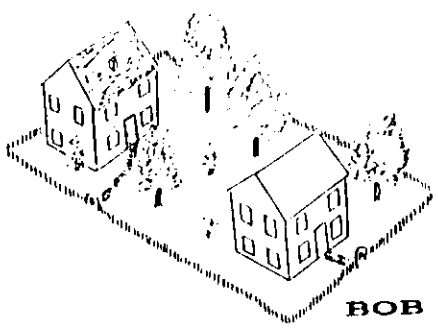
- NON DETERMINISTIC INTERLEAVING

PROBLEMS:

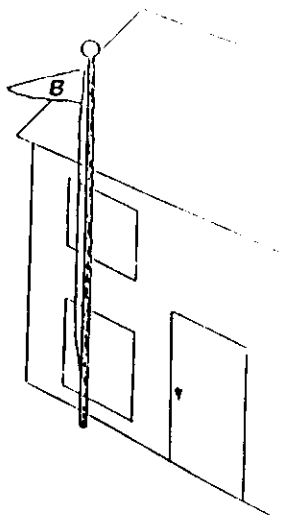
- PROCESSES DESTROY DATA IN A RANDOM WAY
- SYSTEMS CRASH
- RANDOM RESULTS

A CONCURRENCY PROBLEM

ALICE



BOB



HOW TO DESIGN THE PROGRAM?

HOW TO STUDY THE PROGRAM?

- TEST IT?
- PROVE IT?
- SOMETHING ELSE?

DEKKER'S ALGORITHM

```
ONE_TRY := FALSE;
    -- ONE not ask permission to enter

TWO_TRY := FALSE;
    -- TWO not ask permission to enter

TURN := 1;
    -- TURN indicates who is going now
    -- into critical section

procedure ONE ;
procedure TWO ;
begin
  ONE_TRY := FALSE;
  TWO_TRY := FALSE;
  TURN := 1;
  cobegin
    ONE;
    TWO;
  coend;
end;
```

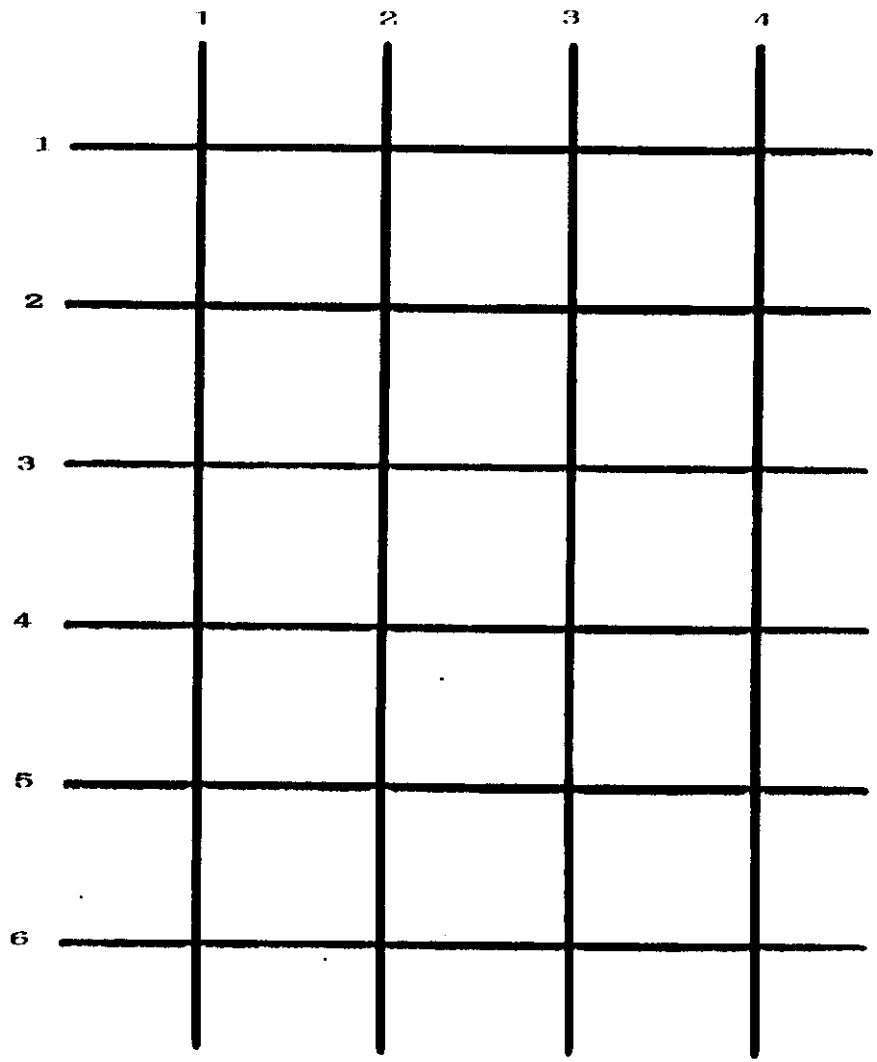
HOW TO SOLVE MUTUAL EXCLUSION PROBLEM IN A SYMMETRIC WAY?

```
procedure ONE is
begin
   loop
      ONE_TRY := TRUE;
      while TWO_TRY loop
         if TURN = 2 then
               ONE_TRY := FALSE;
               while TURN = 2 loop
                     end loop;
               ONE_TRY := TRUE;
         end if;
      end loop;
         -- critical section
         TURN := 2;
         ONE_TRY := FALSE;
         -- non critical section
   end loop;
end ONE;
```



```
procedure TWO is
begin
   loop
      TWO_TRY := TRUE;
      while ONE_TRY loop
         if TURN = 1 then
               TWO_TRY := FALSE;
               while TURN = 1 loop
                     end loop;
               TWO_TRY := TRUE;
         end if;
      end loop;
         -- critical section
         TURN := 1;
         TWO_TRY := FALSE;
         -- non critical section
   end loop;
end TWO;
```

INTERLEAVING METHOD



SEMAPHORES

- ABSTRACT DATA TYPE
- VALUES: 0, 1, 2, ...
- OPERATIONS:

INICIALIZATION (ONLY ONCE)

s := VALUE

WAIT(s):

if s>0 then s := s-1

else execution of task that called WAIT is suspended

SIGNAL(s)

if some task has been suspended by a previous WAIT then wake up it,

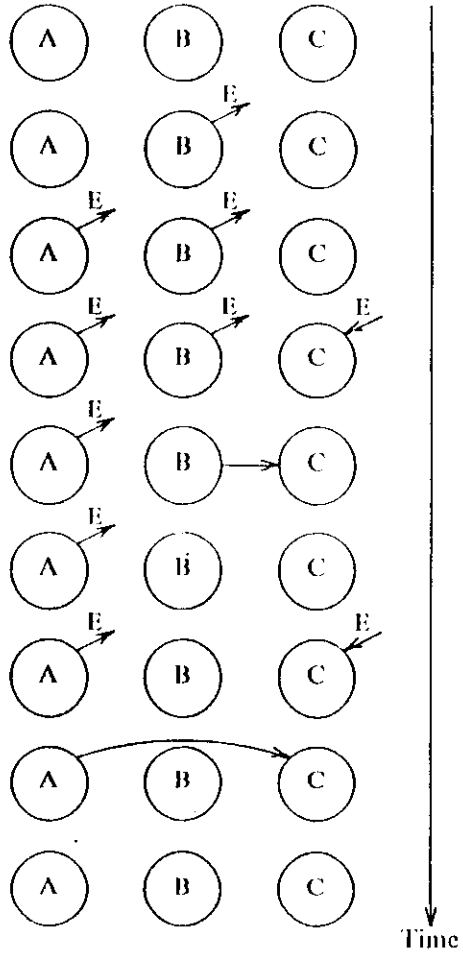
else s := s+1

BOB, ALICE AND SEMAPHORE

```
procedure ALICE is
begin
   loop
      WAIT(s); -- wait for a free yard
      -- dog in yard
      SIGNAL(s); -- signal that the yard
                    is free
   end loop
end ALICE;

procedure BOB is
begin
   loop
         WAIT(s);
         -- dog in yard
         SIGNAL(s);
   end loop;
end BOB;
```


RENDEZVOUS



SEMAPHORES IN ADA

```
task type SEMAPHORE is
   entry WAIT;
   entry SIGNAL;
end SEMAPHORE;

task body SEMAPHORE is
begin
   loop
      accept WAIT;
      accept SIGNAL;
   end loop;
end SEMAPHORE;

procedure
   -- SOMETHING
   SEMAPHORE.WAIT
   -- CRITICAL SECTION
   SEMAPHORE.SIGNAL
   -- OTHER THINGS
end;
```


SELECT CLAUSE

```
select
   accept ONE (...) do
      -- activities
   end ONE;

or
   when FLAG =>
   accept TWO (...) do
      -- activities
   end TWO;

or
   delay 25.0
end select;
```