Lecture 12

Programming models, shared vs distributed memory

Shared memory model

- Processes execute concurrently
- All processes have access to a single shared address space.
- Synchronisation and data exchanges occur implicitly using the shared address space and special data locking operations.
- Existing control flow languages can be augmented to introduce parallelism.

Types of shared memory parallelism

- Data parallelism; execute operations on the members of data structures like arrays and vectors concurrently.
- FORALL/DOALL, loop parallelism; execute iterations of loops concurrently.
- Thread level parallelism; execute blocks of code concurrently.

FORALL

```
FORALL I:= 1 TO 20 DO
   A[I] := I + 1;
END;
.....OR.....
FORALL J:= 2 TO 30 DO
   A[J] := A[ J-1 ];
END
```

Example

```
COBEGIN
        x:= 0; found:= FALSE; a[x]:= 0;
        while ( found != TRUE ) do
                if a[x] = = 2 then
                        found := TRUE;
                end;
        end;
●
        q:= 0;
        if (q<0) then found:= TRUE; end;
●
        q:= -1;
COEND
```


Thread level parallelism

- There are two main structures
  - COBEGIN
  - Parallel procedures

Parallel procedures

- Execute multiple copies of the same code.
- Each copy has a private stack.
- Has the syntax

```
PROCEDURE <procname> ( Id: Integer, ..... );
VAR
        <vardecls>
BEGIN
        <statements>
END <procname>;
```


COBEGIN

- Executes dissimilar code blocks in parallel.
- Operates within the context of the parent process.
- Has the syntax,

```
COBEGIN
        block 1
●
        block 2
●
        block 3
COEND
```

Message passing example

```
PHIL =
*[ THINK;
    ROOM!enter( );
    fork(i)!pickup( ); fork((i+1) mod 5)!pickup( );
    EAT;
    fork(i)!putdown( ); fork((i+1) mod 5)!putdown( );
    ROOM!exit( );
 ]
```


Example

```
PROCEDURE  KParFind ( Id: INTEGER; VAR Image : IntMatrix;);
VAR
    Row, Col, Index: INTEGER;

BEGIN (* KParFind *)
    Row:= Id+1;

    (* Wait to begin *)
    FOR Index:= 1 TO Id DO
        SYNC;
    END;

    WHILE Row < MaxRow DO
        FOR Col:= 2 TO MaxCol DO
            Image[Row, Col]:= Image[Row, Col] * Mask( Image, Row, Col );
            SYNC;
        END;
        Row:= Row + Degree;
    END;
END KParFind;
```


Message passing example

```
ROOM =  occupancy: integer; occupancy:= 0;
*[ (i:0..4)phil(i)?enter( ) -> occupancy++;
 [](i:0..4)phil(i)?exit( ) -> occupancy - -;
 ]
FORK =
*[phil(i)?pickup( ) -> phil(i)?putdown( );
 []phil((i-1)mod 5)?pickup( ) -> phil((i-1)mod 5)?putdown( )
 ]

[room::ROOM||fork(i:0..4)::FORK||phil(i:0..4)::PHIL].
```


Distributed memory model

- Each unit of execution operates within a private address space.
- Communication with other units of execution is conducted through an external interface. Flavours include
  - Message Passing
  - Remote Procedure Calls (RPC)
  - Object/Actor computation.