# Case study 2: Tuple space (concurrent decomposition)

- A *tuple space* is a distributed set of tuples, (terms with a key and zero or more arguments), which form the basis for the Linda parallel programming model.

- *Tuple space* supports a small number of asynchronous operations:

  – **insert (`out`)**

  – **blocking read (`rd`)**

  – **nonblocking read (`rdp`)**

  – **blocking read and delete (`in`)**

  – **nonblocking read and delete (`inp`)**

  – **update by read+delete followed by insertion of modified object**

**(fig 4.7)**

# Application of tuple space in database search

- A database search application can be constructed by concurrent composition of a *tuple space module*, a *manager task*, and multiple *worker tasks*.

**(fig 4.8)**

- Implementation strategies: central server or multiple server tasks

**(fig 4.9)**

# Master and Worker tasks in database search

Pseudocode for master and worker tasks in a tuple-space solution to the database search problem (e.g. finding the max values)

```
procedure manager
begin
  count = 0
  until end-of-file do
    read datum from file
    OUT("datum",datum)      ! Put data into tuple space
    count = count+1
  enddo
  best = 0.0
  for i = 1 to count
    IN("score",value)       ! Retrieve results from workers
    if value > best then best = value
  endfor
  for i = 1 to numworkers
    OUT("datum","stop")     ! Signal workers to stop
  endfor
end


procedure worker
begin
  IN("datum",datum)
  until datum = "stop" do
    value = compare(datum,target)
    OUT("score",value)
    IN("datum",datum)
  enddo
end
```