Université Toulouse III – Paul Sabatier
118 route de Narbonne
31062 Toulouse cedex 9

Practical session 4 – Hoare Monitors

Order processing

On the occasion of the Christmas holidays, your favorite store has decided to make available to its faithful customers, a certain number (NB_COUNTERS) of special counters to take care of their orders. The principle of operation of these counters is as follows:

- A customer waits for a free counter, places his order there, and then waits for it to be ready.
- The processing of an order consists of a number (NB_STEPS) of successive steps and specialized employees are assigned to each of these steps. Step i of the order processing can only start when the employee in charge of the previous step (i-1) has finished his task.
- When the order has been processed, the customer leaves the counter, fully satisfied with this new service.

For example, in practice, the store provides three counters for its customers and the processing of an order consists of four steps: (1) listing the ordered items, (2) picking up the ordered items, (3) wrapping them with gift wrap, and (4) charging the customer.

Problem

We consider two types of processes: Customer and Employee.

- A Customer process places an order at a free counter and waits for this order to be processed before leaving this counter. At most NB_COUNTERS orders can be processed in parallel.
- An Employee process, specialized in step i, takes charge of the first pending order of this step, completes its part of the work before returning the processed order to its original counter. He can then take over a new order. When the last step has been executed on an order, the Customer process that formulated this order can resume its execution.

The following types are assumed to be defined:

- NumberStep : which indicates an integer ranging between 0 and NB_STEPS
- NumberCounters : which indicates an integer between 1 and NB_COUNTERS
- Order : which indicates the order established by a customer

It is also assumed that the subprogram :

- applyStep (NumeroStep numStep, Order *anOrder);

which can be used by an Employe process in order to accomplish its part of the work (i.e. the numEtape step) on a given order and thus make a modification to it.

We propose to synchronize, using a Hoare monitor named GererCmdes, Customer processes and Employee processes so that orders are processed as efficiently as possible.

Specification of this monitor is as follows :

```
Moniteur OrderManagement {
        void command (void) ;          // Used by client
                                       // Used by employees
        void startStep (NumberStep step, Command *cmd, NumberCounters *counter) ;
        void finishStep (Command cmd, NumberCounters counter) ;
}
```

The behavior of the Customer and Employee processes is as follows:

Process Client {
// Go to the store
OrderManagement.command()
// Go home satisfied
}
	Processus Employee (NumberStep myStep) {
	while (True) {
	OrderManagement.startStep(myStep, &cmdToTreat, &counter) ;
	applyStep(myStep, cmdToTreat) ;
	OrderManagement.finishStep (cmdToTreat, counter) ;
	}
	}

Questions :

- Specify the blocking and waking conditions of a Client process and a Employee process.
- Deduce the state variables and the “condition” variables used in the monitor.
- Give the “code” (algorithm) of the monitor.