

Tuto N°. Operating Systems

Process synchronization: Monitors vs. Semaphores

Exercise 1: University Restaurant

Consider a university restaurant organized in such a way that each student respects the following protocol:

DER: Request to enter the restaurant: {*Request a tray and a chair*}

<Eat>

FSR: Finish and exit the restaurant: {*Release a tray and a chair*}

Initially available resources: there are **Pmax** trays and **Cmax** chairs with **Pmax > Cmax**.

-----***-----

I. Solution analysis

University Restaurant

Student i:

- Entrance to the restaurant

" Eat "

-Exit the restaurant

1.The synchronization points:

- **Entry to the restaurant:**

- A. Request a tray

- B. Request a chair

Eat

- **Exit the restaurant:**

- A. Release the chair

- B. Release the tray

2.Synchronization constraints:

1. The number of chairs is limited by Cmax.

2. The number of trays is limited by Pmax.

3. Pmax > Cmax.

Student program:

```

begin
    entrer_rest
    Eat
    sortie_rest
end ;

```

A) Proposed solution using semaphores:

- Use two synchronization semaphores:
- Splat: Semaphore initialized to (**Pmax**). //To limit access to trays.
- Scha: Semaphore initialized to (**Cmax**). //To limit access to chairs.
- Initialize the values of two integer variables **Pmax** and **Cmax**:

Student program:	Procedur entrer_rest	Procedur sortie_rest
<pre> begin entrer_rest Eat sortie_rest end ; </pre>	<pre> begin P(Splat) ; P(Scha) ; end ; </pre>	<pre> begin V(Scha) ; V(Splat) ; end ; </pre>

B) Proposed solution using monitors:

- Using a monitor "Restau" with two procedures "entrer_rest & sortie_rest", and two conditions to limit access to resources (*trays and chairs*).

Student program:	The monitor
<pre> begin call Restau.entrer_rest Eat call Restau.sortie_rest end ; </pre>	<pre> Restau : Monitor; Var Npoc , Ncoc : integer; Cond p , Cond c : condition; Initialisation: Npoc :=0; Ncoc:=0; Begin Procedur entrer_rest; Begin If Npoc=Pmax then Cond p.wait; Npoc:=Npoc+1; If Ncoc=Cmax then Cond c.wait; Ncoc:=Ncoc+1; End DER; Procedur sortie_rest; Begin Npoc:=Npoc-1; Ncoc:=Ncoc-1; Cond p.signal; Cond c.signal; End ; End FSR; </pre>