

TD - Systèmes concurrents

Édouard Lumet - IR2020

Moniteurs ([Audio](#))

Titre

- barbier
- lecteurs / rédacteurs
- interface | prédicats d'acceptation
 - DL() | pas d'écriture en cours
 - TL() | N/A
 - DE() | pas d'écriture et de lecture en cours et pas de lecture en attente
 - TE() | N/A

Variables

```
nl: entier
ecr: booléen
lectureEnAttente: booléen
```

Prédicats

```
DL() : ¬ecr
DE() : ¬ecr ∧ nl = 0 ∧ ¬lectureEnAttente
invariant : nl ≥ 0 ∧ ecr ∈ {V,F} ∧ ¬(ecr ∧ nl > 0) ∧ (lectureEnAttente ∧ ¬ecr)
           ou ¬(ecr ∧ nl > 0)
```

Variables de condition

```
LP (lecture possible, associée à DL)
EP (écriture possible, associée à DE)
```

Implantation de DL()

```

si ¬(¬ecr) alors
    LP.attendre()
Fsi {¬ecr}
nl++
{¬ecr ∧ nl>0}
LP.signaler()    {réveil en chaîne}

```

Implantation de TL()

```

nl--
{¬ecr ∧ nl≥0}
\LP.signaler\Impossible d'avoir lecture en cours et lectures en attente
si nl=0 ∧ LP.Vide() alors
    EP.signaler()
Fsi

```

Implantation de DE()

```

si ¬(¬ecr ∧ nl=0 ∧ \¬lectureEnAttente\LP.Vide()) alors
    EP.attendre()
Fsi {¬ecr ∧ nl=0 ∧ LP.Vide()}
ecr := V
{ecr ∧ nl=0 ∧ LP.Vide()}
{rien à signaler}

```

Implantation de TE()

```

ecr := F
{¬ecr ∧ nl=0}
si LP.Vide() alors
    EP.signaler()
sinon
    LP.signaler()
Fsi

```

FIFO

On sert les requêtes par ordre chronologique d'arrivée

Prédicats

DL() : pas d'écriture en cours et personne n'attend ~ file d'attente vide -> implantée par la file associée à une variable de condition (\$accès)

DE() :pas d'écriture en cours et pas de lecture en cours :

```
invariant :  $\neg(nl > 0 \wedge ecr)$ 
```

Implantation de DL()

```
si  $\neg(\neg ecr \wedge \text{accès.Vide}() \wedge \text{sas.Vide}())$  alors
    accès.attendre()
Fsi { $\neg ecr \wedge$  pas de prédécesseurs dans la file d'attente}
nl++
accès.signaler() {réveil en chaîne}
```

Implantation de TL()

```
nl--
{ $\neg ecr$ }
si nl=0 alors
    si sas.Vide() alors
        accès.signaler()
    sinon
        sas.signaler()
Fsi
```

Implantation de TE()

```
ecr := F
accès.signaler()
```

Implantation de DE()

```
si  $\neg(\neg ecr \wedge nl=0)$  alors
    accès.attendre()
Fsi { $\neg ecr \wedge nl=0$ }
si nl>0 alors // vérification réveil en chaîne
    sas.attendre() // pas accès.attendre() car sinon retour en fin de file
Fsi
ecr := V
```