## Variante 1

### (1) Moniteur

Moniteur lecteur\_redacteur [

public :

void debut\_lire();

void fin\_lire();

void debut\_ecrire();

void fin\_ecrire();

private:

// Variable d’état et variable condition.

};

lecteur\_redacteur lr;

### (2) Processus

Processus Lecteur() {

while(true) {

lr.debut\_lire();

// je peux lire → je lis

lr.fin\_lire();

}

};

Processus Redacteur() {

while(true) {

// je prepare mes données

lr.debut\_ecrire();

// je peux ecrire

lr.fin\_ecrire();

}

};

### (3) Condition de blocage, déblocage d’un lecteur et d’un rédacteur.

* Un lecteur est bloqué dans debut\_lire si un rédacteur écrit
* Le lecteur est réveillé par un rédacteur ayant fini d’écrire si plus personne ne veut écrire dans fin\_ecrire
* Un rédacteur est bloqué dans debut\_ecrire si au moins lecteur lit ou si un rédacteur écrit.
* Un rédacteur est réveillé par un précédent rédacteur ayant fini d’écrire ou le dernier lecteur ayant fini de lire.

### (4) Variables.

bool isWriting = False;

unsignedInt nbLecteur = 0;

Condition lire;

Condition ecrire;

### (5) function V1 (fait par un eleve)

void debut\_lire() {

if(isWriting)

lire.write();

bnLecteur ++;

}

void fin\_lire() {

nbLecteur --;

if(nbLecteur == 0)

ecrire.signal();

}

void debut\_ecrire() {

if(isWriting || bnLecteur > 0)

ecrire.wait();

isWriting = True;

}

void fin\_ecrire() {

isWriting = False;

if(ecrire.is\_empty()) {

while(!lire.is\_empty()) {

lire.signal();

}

} else {

ecrire.signal();

}

}

void initialiser() {

isWriting = False;

nbLecteur = 0;

}

### (5bis) function V1 (fait par le prof)

void debut\_lire() {

if(isWriting)

lire.wait();

bnLecteur ++;

lire.signal();

}

void fin\_lire() {

nbLecteur --;

if(nbLecteur == 0)

ecrire.signal();

}

void debut\_ecrire() {

if(isWriting || bnLecteur > 0)

ecrire.wait();

isWriting = True;

}

void fin\_ecrire() {

isWriting = False;

if(ecrire.is\_empty()) {

lire.signal();

} else {

ecrire.signal();

}

}

void initialiser() {

isWriting = False;

nbLecteur = 0;

}

V1: Famine

* des lecteurs:

un rédacteur demande à écrire avant que le rédacteur précédent ait fini d’écrire (tout le temps)

* rédacteur:

un lecteur demande à lire avant que le dernier lecteur ait fini de lire.

## Variante 2

### (3) Condition de blocage, déblocage d’un lecteur et d’un rédacteur.

* Un lecteur est bloqué dans debut\_lire si un rédacteur écrit ou au moins un rédacteur attend.

### (5) Function

void debut\_lire() {

if(isWriting || !ecrire.is\_empty())

lire.wait();

bnLecteur ++;

lire.signal();

}

Plus de famine des rédacteurs.

## 

## Variante 3

### (3) Condition de blocage, déblocage d’un lecteur et d’un rédacteur.

* Un rédacteur est réveillé par le rédacteur précédent qui a fini d’écrire si aucun lecteur n’attend pour lire ou par le dernier lecteur qui a fini de lire.

Plus de famine.