struct semaphore CanWrite, CanRead;
int AW, AR, WW, WR=0;

reader

if (AW+WW == 0) {
    signal (CanRead);
    AR = AR + 1;
}
else  WR = WR + 1;

wait (CanRead);
:
read
:

AR = AR – 1;
if (((AR==0) && (WW>0)) {
    signal (CanWrite);
    AW = AW + 1;
    WW = WW – 1;
}

}

writer

if (AW+AR+WW == 0) {
    signal (CanWrite);
    AW = AW + 1;
}
else  WW = WW + 1;

wait (CanWrite);
:
write
:

AW = AW – 1;
if (WW>0) {
    signal (CanWrite);
    AW = AW + 1;
    WW = WW – 1;
}
else while (WR > 0) {
    signal(CanRead);
    AR = AR +1;
    WR = WR – 1;
}
struct semaphore CanWrite, CanRead;
int AW, AR, WW, WR=0;
struct semaphore mutex;

**reader**

wait( (mutex);
if (AW+WW == 0) {
    signal (CanRead);
    AR = AR + 1;
}
else  WR = WR + 1;
signal (mutex);
wait (CanRead);
:
read
:
wait (mutex);
AR = AR – 1;
if (((AR==0) && (WW>0)) {
    signal (CanWrite);
    AW = AW + 1;
    WW = WW – 1;
}
signal (mutex);

**writer**

wait( (mutex);
if (AW+AR+WW == 0)
{
    signal (CanWrite);
    AW = AW + 1;
}
else  WW = WW + 1;
signal (mutex);
wait (CanWrite);
:
write
:
wait (mutex);
AW = AW – 1;
if (WW>0) {
    signal (CanWrite);
    AW = AW + 1;
    WW = WW – 1;
}
else while (WR > 0) {
    signal(CanRead);
    AR = AR +1;
    WR = WR – 1;
}
signal (mutex);