BACI Project for an OS Course

This is problem 6.16 on page 213 of the textbook, made specific to Ben-Ari Concurrent Pascal. Write a program containing a monitor **AlarmClock**. The monitor must have an integer variable **theClock** (initialization to zero) and two procedures:

- PROCEDURE **Tick**;
  increments **theClock** each time that it is called. It can do other things, like SIGNALC, if needed, but no WAITCs.
- PROCEDURE **Alarm**(**id** : INTEGER; **delta** : INTEGER);
  blocks caller having identifier **id** for at least **delta** ticks of **theClock**.

You may endow the monitor with any other variables that it needs. The monitor should be able to accommodate up to 5 simultaneous alarms.

The main program should have two procedures outside of the monitor:

- PROCEDURE **Ticker**;
  calls **Tick** in a repeat-forever loop.
- PROCEDURE **Thread**(**id** : INTEGER; **myDelta** : INTEGER);
  calls **Alarm** in a repeat-forever loop.

The main program cannot have any variables. The main program should start 4 concurrent threads: **Thread(100,3)**, **Thread(200,7)**, **Thread(300,10)**, and **Ticker**. Output should occur for each of the following events:

1. At each clock tick, show the values of **theClock**.
2. When any thread either waits or reawakens from a wait, show the id of the thread, the name of the condition on which it is waiting, and the time (**theClock**) at which the wait occurs.
3. When a condition is signalled, show the name of the condition being signalled and the time at which the signal occurs.

Hand in a listing of the program and output showing each **Thread** copy enduring at least one alarm cycle. You may find the **tee** command helpful in this regard.