Lista 5 - Capítulo 7 - Passagens de Mensagens entre Processos

7.2 Consider a filter process **Partition** having the following specifications. **Partition** receives unsorted values from one input channel **in** and sends the values it receives to one of two output channels, **out1** or **out2**. **Partition** uses the first value **v** it receives to partition the input values into two sets. It sends all values less than or equal to **v** to **out1**; it sends all values greater than **v** to **out2**. Finally, **Partition** sends **v** to **out1** and sends a sentinel **EOS** to both **out1** and **out2**. The end of the input stream is marked by a sentinel **EOS**.

(a) Develop an implementation of **Partition**. First give predicates specifying the contents of the channels, then develop the body of **Partition**.

(b) Show how to construct a sorting network out of **Partition** processes. Assume that there are **n** input values and that **n** is a power of **2**. In the worst case, how big does the network have to be? (This is not a good way to do sorting; this is just an exercise!)

7.5 Consider the dining philosophers problem defined in Section 4.3. Develop a server process to synchronize the actions of the philosophers. Show the client (philosopher) interface to the server. The processes should interact using asynchronous message passing.

7.6 Consider the readers/writers problem defined in Section 4.4. Develop a server process to implement the database. Show the reader and writer interfaces to the server. The processes should interact using asynchronous message passing.

7.10 Suppose a computer center has two printers, A and B, that are similar but not identical. Three kinds of client processes use the printers: those that must use A, those that must use B, and those that can use either A or B.

Develop code that each kind of client executes to request and release a printer, and develop a server process to allocate the printers. Your solution should be fair assuming that a client using a printer eventually releases it.

7.13 Consider the self-scheduling disk driver process in Figure 7.9.

(a) Modify the process to use the CSCAN scheduling strategy that was employed in the monitor in Figure 5.13.

(b) Modify the self-scheduling disk driver in Figure 7.9 to use the SCAN (elevator) scheduling strategy described at the start of Section 5.3.

7.16 You are given three processes `F`, `G`, and `H`, each having a local array of integers. All three arrays are sorted in nondecreasing order. At least one value is in all three arrays. Develop a program in which the three processes interact until each has determined the smallest common value. Give key assertions in each process. Messages should contain only one value at a time.