

CHAPTER 5: INTERTASK COMMUNICATIONS AND CONTROL

The next example provides for two very simple interactive tasks, each sending messages to the other. The messages are input via the keyboards of each task, and appear on the screen of the other task. This is done using separate windows controlled by separate tasks running concurrently under the VSE Run-Time Monitor. Figure 5.1 below shows the architecture of this simple example. The implementation follows in Figure 5.2.

The session starts by the user running task 1, which automatically opens a window. Task 1 immediately starts task 2, with a window, and suspends itself. The very first message of the session, 'ASK A QUESTION', is put on the screen of task 2 by initialization. From then on, the conversation proceeds with the keyboard entry being put into CONVERSATION_BUFFER, an intertask resource. The task that accepts input from the keyboard then resumes the other task and suspends itself. When a task is resumed, it displays the message in CONVERSATION_BUFFER upon the screen, accepts the next input from the keyboard, putting it into the CONVERSATION_BUFFER, resumes the other task and suspends itself. This continues until one of the keyboard entries is STOP.

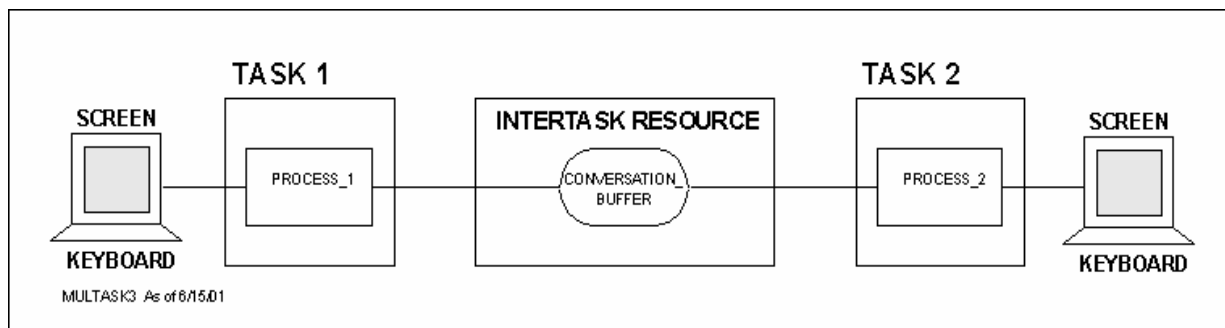


Figure 5.1 Architecture of real-time intertask communications example.

```

                                INTERTASK RESOURCE: CONVERSATION_BUFFER

CONVERSATION_DATA
1      ANSWER                    CHAR 4
                                VALUE 'STOP'
                                ALIAS STOP
1      REST                      CHAR 60

                                TASK 1
PROCESS_1

PROCESS_1
START TASK_2 WITH WINDOW
SUSPEND TASK_1
EXECUTE ANSWER_A_QUESTION
UNTIL ANSWER IS STOP
TERMINATE THIS TASK

ANSWER_A_QUESTION
DISPLAY CONVERSATION_BUFFER
ACCEPT CONVERSATION_BUFFER
RESUME TASK_2
SUSPEND TASK_1

TASK CONTROL SPECIFICATION : TASK_1

*CONTROL SECTION
TITLE, EXAMPLE OF INTER-TASK COMMUNICATIONS & CONTROL
LEAD_PROCESS = PROCESS_1
*END

                                TASK 2
PROCESS_2

PROCESS_2
MOVE 'ASK A QUESTION' TO CONVERSATION_BUFFER
EXECUTE ASK_A_QUESTION
UNTIL ANSWER IS STOP
RESUME TASK_1

ASK_A_QUESTION
DISPLAY CONVERSATION_BUFFER
ACCEPT CONVERSATION_BUFFER
RESUME TASK_1
SUSPEND TASK_2

TASK CONTROL SPECIFICATION : TASK_2

*CONTROL SECTION
TITLE, EXAMPLE OF INTER-TASK COMMUNICATIONS & CONTROL
LEAD_PROCESS = PROCESS_2
*END

```

Figure 5.2. Elements of the AVERAGE Task.

Anyone who has worked with intertask communications and control in UNIX, referred to as Inter-Process Communication (IPC), will testify to the level of difficulty involved in creating the little example above. All of the effort of setting up and managing shared memory control blocks, shared memory areas, and the difficulties of putting processes to sleep and sending signals to wake them up is done for the user, behind the scenes, by VSE. These are powerful features for real-time communications and control applications.