Published: 11/25/66

Identification

Restart
R. L. Rappaport

Purpose

Entry point restart in the Traffic Controller allows running processes, which are giving up control of a processor, to schedule future processor usage.

Preface

The description of restart that follows is divided into two sections. The first section presents the basic outline of the subroutine. This would be an adequate description if it could be assumed that execution of the subroutine would take place while:

1) The processor was completely masked against interrupts.

2) A global interlock was on which denied access to the Process Exchange to all processes except the one in which this subroutine is currently executing.

The final section is a complete specification that describes the steps that must be taken to allow more than one process to concurrently execute in the Traffic Controller.

Basic outline

Restart is called by the Process Interrupt Handler upon receipt of a timer runout interrupt or a pre-emption interrupt. The calling sequence is simply:

```
call restart;
```

and the stack used on this call is the Process' Concealed Stack.

Restart is actually a very simple procedure:

1. It calls the scheduler to place this process on the ready list. (See section BJ.4.00)

2. It calls getwork (Section BJ.4.02) to give control of the processor to some other process. It should be noted that a return from getwork is not experienced until some time in the future when another process switches control of a processor to our original process. The processor executing the return from getwork need not be the same as the calling one.

3. Upon return from getwork, restart returns to its caller.

Complete Specification of Restart

With several processes possibly executing in the Process Exchange concurrently, steps must be taken to coordinate their actions. Restart makes one small contribution to this coordination effort. Upon entry restart sets on the intermediate-state switch of the calling process. This switch exists as a data item in the Active Process Table entry of the process. For a full understanding of the issues involved in this step it is important to notice several things. When return is experienced from the scheduler, the process although still executing appears to be in the "ready" state because the process' Active Process Table entry indicates this to be the case. However, disaster would follow if another process should attempt to switch control of another processor to this process, while this process is still executing. The disaster would not be caused by the fact that two processors with identical descriptor segment base registers were executing simultaneously, but because each of these processors would be referencing and altering the same stack segment without the other being aware of it. In order to avoid this possibility, the process intermediate-state switch is provided. If this switch is on, it indicates that the process may be executing in the Process Exchange, even though its "state", as defined in its Active Process Table entry, may not indicate this. The intermediate-state switch is reset in swap-dbr by the process to whom control of the processor is given. (See Section BJ.5.01) For a complete discussion of coordination of Process Exchange modules see Section BJ.6.

Figure 1 is a complete flow diagram of restart.

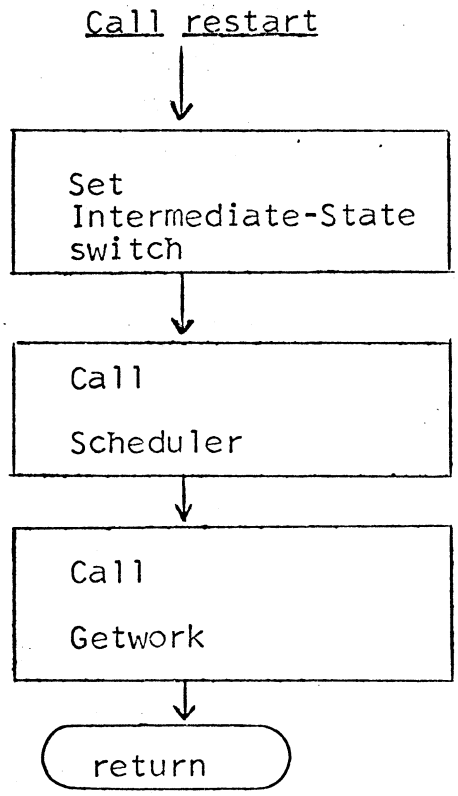

Figure 1

Complete flow diagram of restart.