Draft 6.10.68

Identification

The Scheduler

Robert L. Rappaport, Michael J. Spier

Purpose

This section describes the multi-level scheduling algorithm which is used by the Traffic Controller in order to multiplex the available processor-time among ready processes.

Background

The reader is referred to the paper "An Experimental Time Sharing System" by F. J. Corbató, M. Merwin-Daggett and R. C. Daley which was presented at the 1962 Spring Joint Computer Conference and which describes the F. J. Corbató multi-level scheduling algorithm.

Introduction

The Traffic Controller maintains in the Active Process Table (APT) a threaded list of all ready processes, named the ready-list; it is a linear thread which goes through the APT and which is broken into subthreads, or queues. Typical of the ready list (see BJ.1.2) is that it can be accessed sequentially as a whole (when selecting the next process to run), or partially by means of direct access to a given queue (when putting a process on the ready list.) The number of queues in the ready list is fixed at system-initialization time; queues are numbered from 1 (highest priority) to q (number of queues, lowest priority) and are accessed directly by their number. Associated with each process in the system is a level number l which corresponds to the ready list queue into which this process'APT entry may currently be threaded.

Also associated with each process (and kept in the Process' Definition Segment (PDS)) are two level numbers, named lowest-level and highest-level, ~~and~~ which delimit the ~~highest~~ lowest- and highest-priority level numbers this process can assume, and a value named time-unit and which is used in the computation of the process' time allotment.

The scheduling algorithm always increments a process' current level number by one, putting it into a lower-priority queue, but not below its lowest-level mark. If it is called with an "interaction-sw" set to "on" it knows that the process is interacting with a human being and is blocking itself awaiting human response. The scheduler then sets the process current level number to highest-level.

The computed time allotment is a function of the process' current level number and its time-unit.

The scheduling algorithm

Following is the scheduling algorithm, in PL/1 language. "level" stands for the process' current level number, all other symbols have been mentioned above:

```
    if level < highest_level then level = highest_level - 1;
    if interaction_sw then level = highest_level - 1;
    level = level + 1;
    if level > lowest_level then level = lowest_level ;
    time_allotment = time_unit * (2 ** level) ;
```

2
0

Implementation

The scheduler is invoked in two places only, subroutine restart and subroutine block wheb block is called with an interaction-sw = "on".

Therefore, for reasons of efficiency, the scheduler is coded in-line and does not include the second IF statement when invoked in restart. When a process is awakened for the very first time, and the awakening process does not know its level-boundaries nor its time-unit, the process is affected a current level number which is a system constant and a time allotment which is barely sufficient to carry the process out of the Traffic Controller. It then gets a timer runout interrupt and reschedules itself properly.

Certain level numbers are reserved. Level number 1 (the highest-priority) is reserved for system daemon processes (i.e. the Traffic Controller Daemon Process). Level number q (the lowest priority) is reserved exclusively for the idle processes; these processes never block themselves, they always call restart and thread themselves onto the ready list. Consequently, the ready list is never empty.