Real Time systems2

Using Primary and alternative tasks

Using Primary and alternative tasks

- We Have assumed that there must always be sufficient time for critical tasks to execute.
- In order to ensure that critical tasks will complete before their deadline, we carry out a scheduling that assumes that each critical task will run to its worst-case time.
- Quit often, the worst-case execution time of such tasks is much greater than the average -case execution time.

Using Primary and alternative tasks

This results in much more time being scheduled for the tasks than is really needed.

One way of retaining a high utilization of the hardware is to reclaim for less critical functions the time left unused when the critical tasks do not need all their scheduled time.

Using Primary and alternative tasks

- Now we have another approach: !!!
- Suppose that for each critical task, we have two versions:
- a primary and alternative.
- Completing either the primary or alternative version successfully results in the critical task being executed.

Using Primary and alternative tasks

- The alternative is a “bare-bones” version that provides service that is just acceptable.
- While the primary may be capable of providing better-quality service.

- The alternative version has a most smaller worst execution time than the primary.

- Since only one of these versions has to execute in time to ensure acceptable service, we can avoid having to pre-allocate the primary for its worst-case time.

Using Primary and alternative tasks

- Example :
- Consider we have only one task:

	Primary	alternative
Worst-case run time	20	5
Average run time	7	4
Period	15	15

- If only the primary version were available, this task would impossible to be scheduled !!.
- Since now we have alternative.

Using Primary and alternative tasks

- We allow 10 time units for the primary version to run in each period of 15 time units. We call this the run time limit of primary version. Much of the time (since the average run time is only 7), the primary version will have completed by that time.

- And we reclaim the time beyond the completion time for other activities.
- Now, if the primary runs for more than 10, we abort it and start up the alternative task.
- Acceptable output.

Primary										Alternative				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Using Primary and alternative tasks

The algorithm:

- Assume that a set of tasks is periodic.
- The periods are in the set {Pm,2Pm, 2^2Pm, 2^3Pm,…., 2^iPm}.

- A task is said to be of level-i if its period is 2^iPm, i=0,1,2,….

- Now we present two uniprocessor scheduling algorithms:
- 1. Generating the initial schedule.
- 2. for reclaiming unused time from the initial schedule.

Using Primary and alternative tasks

- The initial schedule is generated as follows:
- 1. We schedule all level-0 tasks over an interval Pm, ensuring that all alternative versions of such tasks are scheduled.
- 2. Schedule the maximum number of primary versions that will fit in the remaining time.
- Important note: The alternative version of a task is never scheduled to run before its primary.
- This we call it Schedule level-0 S0.

Using Primary and alternative tasks

- 3. Concatenate two S0 schedules to form one schedule of length 2Pm.
- 4. Schedule all level-1 tasks in the following manner:
- First, schedule the alternative versions. If there is insufficient space in the schedule to fit all the alternatives, drop some of the primary versions of level-0 tasks, as necessary.
- If primary versions have to be dropped, drop the ones that have the longest run-time limits.

Using Primary and alternative tasks

- The idea is to drop a few of them as possible).

- 5. Once all the level-1 alternatives have been scheduled, see if any of the level-1 primaries can be scheduled in the space available. If they fit do so.

- 6. Check to see if any as-yet-unscheduled level-1 primaries have a lower run time than any primary already scheduled??
- if so, drop the already-scheduled primary with the longest run-time limit and replace it with level-1 primary.

Using Primary and alternative tasks

- 7. Concatenate two copies of the resultant schedule together to form a schedule of length 2^2Pm.

- 8. schedule level-2 tasks in the same way- drop level-1 or level-0 primaries as necessary to schedule level-2 alternatives.

- Drop the ones with the longest run-time limit first.

9. Continue until at least all alternative versions are scheduled.

Using Primary and alternative tasks

- Example:
- Consider the following task set of five tasks. Pr(i) is the worst-case run time. and Al(i) is the alternative version of T(i). P(i) is the period.

Task	T1	T2	T3	T4	T5
Pr(i)	10	10	15	10	5
Al(i)	3	2	1	7	4
P(i)	20	20	20	40	40

- لاحظوا الفترات للمهام T1, T2, T3 هي 20 بينما لT4, T5 هي 40. و نلاحظ أن 40 هي ضعف ال 20 اي لدينا مستويين

Using Primary and alternative tasks

- Level 0 task are T1, T2, T3.
- Level-1 tasks are T4 and T5.
- Now we start scheduling level-0
- First we schedule the alternative version.

Al(1)			Alt(2)		Al(3)													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

- We have only 14 time units left.
- We can schedule Pr(1) or Pr(2). We choose Pr(1) for example.

Using Primary and alternative/Example

- Note: this solution and you may generate another correct solution.
- Note: Never put primary after alternative.
- Now S0 will become.

Pr(1)										Al(1)			Al(2)		Al(3)	idle	idle	idle	idle
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

- Now we concatenate two copies of S0.

Pr(1,1)										Al(1,1)			Al(2,1)		Al(3,1)				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Pr(1,2)										Al(1,2)			Al(2,2)		Al(3,2)				
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

Using Primary and alternative tasks

- Now we have to schedule at least the alternative version of al(4) and al(5).

- This requires 11 time units we haven't.!!!!
- So we drop Pr(1,2) and add alternative versions of T4 and T5. See below:

Pr(1,1)										Al(1,1)			Al(2,1)		Al(3,1)	Alt(4)			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Al(4)			Pr(5)					idle		Al(1,2)			Al(2,2)		Al(3,2)	Alt(5)			
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

Using Primary and alternative tasks

- HW: Schedule tasks using primary and alternative algorithm.

Task	T1	T2	T3	T4	T5
Pri	2	3	3	7	8
Alti	1	2	2	1	2
Pi	5	10	10	20	20