CSE 3141 Prac #4-5: Analysing Scheduling using the Cheddar Real Time Simulator

Dr Carlo Kopp, PEng,
Lecturer, SCSSE, Clayton.
Email: carlo@mail.csse.monash.edu.au

May 24, 2004

Revision Status:

```
$Id: CSE-3141-Prac-4-5.tex,v 1.2 2004/05/17 15:24:12 carlo Exp carlo $
```

Contents

1 Introduction

A common problem you will encounter in the design of a real time system is that of both designing and validating a scheduling policy for the intended system. Traditionally, for system designs this was done using a pencil and paper chart to map out the behaviour of the scheduling algorithm for a given type of workload.

The reality is that many workloads can be complex in their behaviour and this can cause genuine difficulties especially in validating system timing performance.

This prac will introduce the recently developed `cheddar` simulator, created by the EA 2215 team at the University of Brest in France. This simulator can be used for a range of such tasks and upcoming pracs will involve using it to perform such tasks. It is used by a number of universities in the EU, US and elsewhere.

Students are encouraged to install `cheddar` on their home systems. Binaries and required libraries for Linux and MS Windows are available on the `cheddar` website at:

`http://beru.univ-brest.fr/~singhoff/cheddar/`

The website also contains the documentation and user guide.

The first 2 hours of this prac are intended to familiarise you with the simulator, and allow you to install it under Linux and test it in a supervised environment. Subsequent weeks will involve specific tasks.

The scheduling theory will be covered in upcoming lectures and students are encouraged to read ahead through the lecture notes.

2 Lab Session 1 - Installation and Familiarisation (2 hr)

Download the gzipped tarfile `cheddar-1.3-runtime.tar.gz` into your home directory. Extract the file in your work directory of choice. The file contains Linux binaries, libraries, user guide and example XML work files for the `cheddar` simulator.

To run the `Cheddar` simulator you must configure your environment so the runtime loader can locate the binaries and runtime libraries required, which are not installed on your Linux systems. This is accomplished by running the following commands from `tcsh` or `csh`:

```
setenv LD_LIBRARY_PATH ~MyHome/MyWork/cheddar/lib:$LD_LIBRARY_PATH
setenv PATH ~MyHome/MyWork/cheddar/bin:$PATH
```

If you are running Korn shell or `bash`, then you can achieve the same using:

```
export LD_LIBRARY_PATH=~MyHome/MyWork/cheddar/lib:$LD_LIBRARY_PATH
export PATH=~MyHome/MyWork/cheddar/bin:$PATH
```

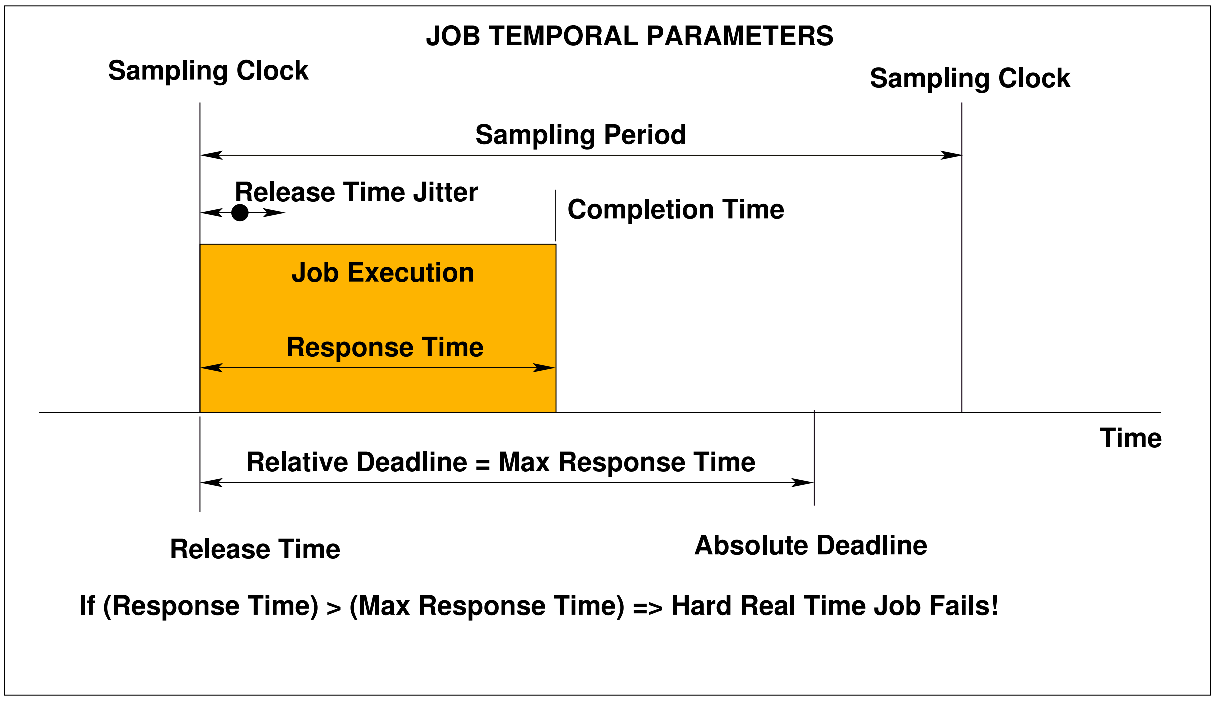
You can now run `cheddar` from the command line.

Once you have a running installation, read the user guide and work through the `.xml` sample files in the `examples` directory tree.

Your assessment will be based on whether you have successfully installed and test run `cheddar` during the available time.

3 Lab Session 2 (2 hr duration)

Your tasks are as follows, for each of these activities:



Activity (0) - Perform all steps.

1. Start Cheddar and open file Prac-4-1.xml.
2. Use the Edit->Update->Update-Tasks menu to view the three tasks defined. These are periodic tasks all of which share a period of 10, deadlines of 10, two have execution times (labelled as 'capacity' in Cheddar) of 1, one an execution time of 2, and all have monotonically increasing priorities from 1 to 3.
3. Use the Edit->Update->Update-Processors menu to view the single processor defined. It uses a 'Highest Priority First' scheduling algorithm (POSIX), and in non-pre-emptive.
4. Run the 'scheduling simulation' from 0 to 30 and record the results on a worksheet. Task execution is drawn in black and task release times marked in red on the time axis.
5. Use the File->Save-as menu to make a copy of this file as MyId-Prac-4-1-0.xml.

Activity (1) - Perform all steps.

1. Make a copy of Prac-4-1.xml and name it MyId-Prac-4-1-1.xml.

2. Experiment by increasing the execution time of all three tasks concurrently in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (c) What are the respective response times for the longest combination of execution times producing a feasible schedule?

Activity (2) - Perform all steps.

1. Make a copy of Prac-4-1.xml and name it MyId-Prac-4-1-2.xml.
2. Modify the period of Task_1 from 10 to a harmonically related value of 20, with a deadline of 20. Then experiment by increasing the execution time of all three tasks concurrently in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (c) What are the respective response times for the longest combination of execution times producing a feasible schedule?

Activity (3) - Perform all steps.

1. Make a copy of Prac-4-1.xml and name it MyId-Prac-4-1-3.xml.
2. Modify the period of Task_1 from 10 to a harmonically related value of 20, with a deadline of 20, and Task_2 to a period and deadline both of 30. Then experiment by increasing the execution time of all three tasks concurrently in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?

(c) What are the respective response times for the longest combination of execution times producing a feasible schedule?

Activity (4) - Perform all steps.

1. Make a copy of Prac-4-1.xml and name it MyId-Prac-4-1-4.xml.
2. Modify the period of Task_1 from 10 to a harmonically related value of 20, with a deadline of 20, and Task_2 to a period and deadline both of 30. Make the scheduling pre-emptive rather than non-pre-emptive as in the previous examples.

   Then experiment by increasing the execution time of all three tasks concurrently in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (c) What are the respective response times for the longest combination of execution times producing a feasible schedule?
   (d) What is the impact of pre-emption on the response times and usable execution times?

4 Lab Session 3 (2 hr duration)

Your tasks are as follows, for each of these activities:

Activity (1) - Perform all steps.

1. Browse the Tasks and Processor configuration of Prac-5-1.xml. Note the deadlines and the use of the Earliest Deadline First (EDF) scheduling algorithm.
2. Using the Prac-5-1.xml simulation, experiment by increasing the execution time of each task in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file as MyId-Prac-5-1-1.xml for submission, containing the first infeasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) Which task fails first and why?
   (c) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (d) What are the respective response times for the longest combination of execution times producing a feasible schedule?

Activity (2) - Perform all steps.

1. Make a copy of Prac-5-1.xml and name it MyId-Prac-5-1-2.xml.
2. Modify the period of Task_1 from 10 to a harmonically related value of 20. The deadlines remain at 8, 9 and 10 respectively. Then experiment by increasing the execution time of each task in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (c) What are the respective response times for the longest combination of execution times producing a feasible schedule?
   (d) Which task fails first and why?

Activity (3) - Perform all steps.

1. Make a copy of Prac-5-1.xml and name it MyId-Prac-5-1-3.xml.

2. Modify the period of Task_1 from 10 to a harmonically related value of 20, with a deadline of 20, and Task_2 to a period of 30 and a deadline of 7. Then experiment by increasing the execution time of each task in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what values of execution time can a feasible schedule not be found?
   (b) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (c) What are the respective response times for the longest combination of execution times producing a feasible schedule?
   (d) What are the most notable differences between this schedule and the previous one?

Activity (4) - Perform all steps.

1. Make a copy of MyId-Prac-5-1-3.xml and name it MyId-Prac-5-1-4.xml. Make sure that Activity (3) was properly completed.
2. Experiment by decrementing the deadline of Task_1 in increments of 1 until a feasible schedule cannot be found (no task can miss its deadline in the scheduling simulation). Once finished, save your file for submission, containing the final feasible schedule.
3. Answer the following questions:
   (a) At what value of Task_1 deadline can a feasible schedule not be found?
   (b) Which task failed and why?
   (c) What is the processor utilisation for the longest combination of execution times producing a feasible schedule?
   (d) What are the respective response times for the longest combination of execution times producing a feasible schedule?

1. You will submit by email all of the files you produce at the end of the lab period. Late submissions will attract a loss of one mark per day.
2. You must demonstrate your work to the demonstrator before the end of the second prac session. If your simulation does not work at all you will be awarded zero marks for this prac. If your analysis has errors or is incomplete, marks will be deducted accordingly.
3. **Cheating, collaborating and plagiarism will attract zero marks.**

5 Lab Session 4 (2 hr duration)

Your tasks are as follows, for each of these activities:

Activity (1) - Perform all steps.

1. .Browse the Tasks and Processor configuration of Prac-5-2-1.xml. Note the release times, deadlines and the use of the Earliest Deadline First (EDF) scheduling algorithm. Copy this file to MyId-Prac-5-2-1.xml.
2. Add a new processor called Processor_MyId, using the Least Slack Time (LST - ' Least Laxity First') scheduling algorithm. Add two tasks called Task_2 and Task_3, identical to the first two tasks, running on Processor_MyId. Run the scheduling simulation and save your result for submission.
3. Answer the following questions:
   (a) Assume each context switch consumes 100 microseconds of CPU time. What is the total context switching time expended for EDF and for LST scheduling, respectively?
   (b) Explain why the LST algorithm produces a different number of context switches to the EDF algorithm.
   (c) What is the effect in terms of response times, for both EDF and LST?
   (d) Given your findings, and this system, where would you employ EDF in preference to LST, and vice versa?

Activity (2) - Perform all steps.

1. Make a copy of MyId-Prac-5-2-1.xml and name it MyId-Prac-5-2-2.xml.
2. Experiment by incrementally increasing the release time of Task_1 / Task_3 from 0. Save the file with the configuration which fits your answers.
3. Answer the following questions:
   (a) At what release times is the context switching overhead minimised?
   (b) Why is this so?

Activity (3) - Perform all steps.

1. Make a copy of Prac-5-2-3.xml and name it MyId-Prac-5-2-3.xml. This simulation represents a system in overload, as the total available processor time is not enough for all three jobs to meet their deadlines.
2. Run the simulation using both the EDF and LST scheduling algorithms, from 0 to 20. Save the run which uses the LST algorithm.

3. Answer the following questions:
   (a) Calculate the miss rates for both algorithms.
   (b) Calculate the peak and arithmetic mean lateness for both algorithms.
   (c) Calculate the peak and arithmetic mean tardiness for both algorithms.
   (d) Calculate the Effective Processor Utilisation (EPU - refer lecture notes on webpage) for both algorithms.
   (e) Using the scheduling algorithm which minimises the miss rate, what is the effect on the jobs which miss their deadline(s) in terms of lateness and tardiness?
   (f) Explain why the EDF algorithm is not favoured for systems prone to overloads.

1. You will submit by email all of the files you produce at the end of the lab period. Late submissions will attract a loss of one mark per day.
2. You must demonstrate your work to the demonstrator before the end of the second prac session. If your simulation does not work at all you will be awarded zero marks for this prac. If your analysis has errors or is incomplete, marks will be deducted accordingly.
3. **Cheating, collaborating and plagiarism will attract zero marks.**