

MIT OpenCourseWare
<http://ocw.mit.edu>

6.033 Computer System Engineering
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

Design Project 1: Frequently Asked Questions

Q: Do I really need to stay within 2,500 words?

A: Yes. We expect you to adhere to the word limit. That said, we're not going to penalize you for being just a few words over the limit -- we're mostly concerned with the quality of your report. If your report contains 2,500 + X words, and you're worried whether X counts as "just a few," you should probably cut some words. ***Remember to print the word count at the end of your final report.***

Q: Do the figure captions count against the word limit?

A: Yes. Don't skimp on captions to lower your word count however; appropriate captions are important for clarity.

Q: Do we have to use the exact API as provided, ie. emit(), body() etc?

A: You should try to design your system to fit with the given API as close as possible. The implementation of functions such as run(), init(), body() and emit() are of course up to you. You may change the arguments to these functions where required by your design, i.e. you may wish to have emit() take an argument specifying the operator ID of the output destination, rather than specifying the output link ID.

Q: Is there a way for the scheduler to predict the CPU execution time of each operator when making scheduling decisions?

A: It is fine to assume that your scheduler can estimate the runtimes of operators when scheduling them, and you may choose to add this information to the dataflow graph description. That said, a design with a scheduler that has no specific knowledge of operator costs can still get a good grade -- be careful to come up with a solid design before adding any additional details.

Q: Do we need to consider variances in parameters such as access frequency and CPU times, and optimize both for average case and worst case executions?

A: Although variances in system characteristics are often an issue in practice, for the purposes of your design and analysis you should assume that any variances are negligible.

Q: Can we assume that a multi-input operator needs to receive data on each input before producing an output?

A: Some multi-input operators may produce an output after receiving only one (or more) inputs. An example of this is the stateless merge operator in Figure 6 of the design project. This is a characteristic of each specific operator. Remember that you may add additional fields to the dataflow graph specification where appropriate, if necessitated by your design.

Q: Is each operator *required* to run in its own thread?

A: No. There is no single correct solution to the design project, and you should not feel constrained by any sample solutions that might be discussed in tutorial.

Q: How should we account for context switch overhead?

A: For simplicity, you may assume that context switches only occur at transitions between operators.

Q: If two operators both feed into one destination operator, and both do so concurrently, will this lead to a race condition?

A: Our model requires that all operators are one-to-one, such that each of the two source operators have separate channels to the destination operator. It's up to you to define the mechanism by which data is transferred between operators, and when you do so you must ensure that the system operates correctly when an operator receives data from two input channels concurrently.

Q: At what time is DP1 due?

A: DP1 is due at the start of Recitation 13.

Q: Does all the input to the graph come from a single source?

A: No, you should assume that dataflow graphs may receive inputs from multiple input operators.

Q: Does pseudocode contribute towards the word count?

A: You may exclude pseudocode from the word count. With that in mind, you should be careful not to use pseudocode in place of an adequate description of your design in the body of your report. The recitation and writing program instructors will be looking to see that your design is well-described in the actual text of your report, and that certain design choices aren't described only in pseudocode.