* Activity
  + scholarly work—exploring a topic, reading relevant literature, writing software, and running experiments—and must include either a substantial amount of programming or a strong theoretical mathematical treaty of issues related to multicore programming.
* Report
  + short paper (6–10 pages) describing the results of the activity
  + make clear what is the problem, the state-of-the art in this research area, a discussion of related work,
  + a clear definition of the contributions of your paper, a detailed discussion of your algorithms and approach and their application, and performed experimental results
  + graded for writing quality as well as for technical content, so attention should be paid to organization, grammar, spelling, and scholarly style.
  + All references used, both text and code, must be properly cited in the bibliography.
  + Your are required to use LaTeX ([http://en.wikipedia.org/wiki/LaTeX (Links to an external site.)](http://en.wikipedia.org/wiki/LaTeX)) to prepare your report.
* Format:
* All papers should follow either the ACM or the IEEE Manuscript Format.
* A sample IEEE template and further formatting instructions are available here:
* <http://www.ieee.org/conferences_events/conferences/publishing/templates.html>
* prepare your paper using LaTeX and that you prepare your references using the companion tool, BibTeX. Exceptions require the instructor’s prior approval and will typically not be allowed.
* Compared to papers produced on a typical word processor, the results from LaTeX are much more professional looking. LaTeX automatically numbers pages and sections, automatically produces a table of contents and list of figures (if desired) and generates page headers and cross-references.
* BibTeX takes information that you put into a bibliographic database and formats it according to commonly-accepted styles. You don’t have to remember whether the article title or journal title should be italicized, or where commas and periods are needed—it does all that for you.
* LaTeX produces professional-quality typeset equations and formulas rather painlessly.

RP opening

* The paper should begin with title, author, and abstract.

RP body

* The body of the paper should be divided into logical sections appropriate to the structure of the material. Each section should have a numbered section heading. Numbered and unnumbered subheadings should be used where appropriate
* Figures and tables should have captions and should be referenced by number. Related work should be cited in the text, and full reference information should appear at the end in a bibliography
* Any material copied verbatim should be enclosed in quotation marks as well as being properly cited.
* the project report should have some ideas of your own to report. It should reach some conclusion, and it should give logical arguments and relevant data to support that conclusion.
* What I do not want in a paper is a simple paraphrasing of somebody else’s work. Quoting other people’s work as a way to make a point is perfectly acceptable, if properly attributed; simply copying their work, attributed or not, is not acceptable.
* I want to read about your ideas, your code, and your conclusions, not somebody else’s. But of course you will rely on other people’s work to support your arguments.

Appendix (a supplement to the main body text)

* briefly list the challenges, the set of completed tasks, and the set of remaining tasks for the project.

Bibliography

* The bibliography should contain full citation information, including author, title, year, and publication data. If the publication is a conference proceedings, then the proceedings title, editor, organization or publisher, etc. should be included. If it’s a book, then the publisher should be included. If it’s a web page, then the URL and sponsoring organization should be mentioned.

Suggestions and hints

* study of the use and practical application of nonblocking data structures in benchmarks and real-world algorithms and applications. Find an algorithm or application that would benefit from the use of a multicore nonblocking data structure (hash table, queue, vector):
* The application of program analysis for automatic parallelization, such as the use of an open compiler infrastructure such as ROSE ([http://www.rosecompiler.org/ (Links to an external site.)](http://www.rosecompiler.org/) ) to translate old code relying on mutual exclusion to lock-free code.
* The use of formal verification and validation tools such as model checkers and theorem provers (such as the SPIN Model Checker, Alloy Anayzer, TLC, and many more) to validate key properties of concurrent algorithms.