CS 321 – File Structures
Syllabus – Fall 2009

INSTRUCTOR: Tom Naps

OFFICE: Halsey 229, phone 424-1388

EMAIL: naps@uwosh.edu

OFFICE HOURS: MWF 3:00-4:00, TueThur 12:30-1:30

EXAMS:
- Friday, October 9th
- Monday, November 23rd,
- Friday, December 18th

REFERENCES:

- Daily class handouts. Organize them, take notes on and about them. Handouts that are not liberally saturated with your own explanatory notes will likely prove useless when you need them most
- Course Web page at `http://csf11.acs.uwosh.edu/cs321`.

Topic Coverage

1. Text vs. Binary Files
2. Sequential File Processing
3. Security: data integrity, authentication, and encryption algorithms
4. Hardware characteristics of secondary storage devices
5. External sorting
6. Direct (random) access file processing
7. Indexing and hashing schemes for randomly accessible data, including
   (a) primary versus secondary index
   (b) multi-level index
   (c) B-trees and B+-trees
   (d) tries
   (e) PATRICIA trees
   (f) extendible hashing
   (g) bitmap index
   (h) kd-trees
8. Buffering
9. Page-replacement algorithms
10. File compression algorithms

Learning Outcomes

With our coverage of these topics, . . .

1. Given the mechanical characteristics of a hard drive, the student will be able to compute the minimum, maximum, and average latencies of file accesses.
2. Given the description of a text- or binary-file processing task, the student will be able to implement a solution for it using the language (e.g., C or Java) primitives for accessing external memory, and taking advantage of buffering whenever possible.
3. Given a maximum RAM capacity and a set of file records stored on disk, the student will be able to sort them according to a given key field using the k-way sort-merge approach, both with and without the replacement-selection algorithm.
4. Given a set of records on file with a primary key and a value of the desired B-tree order, the student will be able to construct the B-tree or B+ tree index of the given file by repeatedly applying the two-pass insertion algorithm.
5. Given a B-tree index and a key value, the student will be able to delete the entry in the B-tree corresponding to the file record with the given key value.
6. Given the description of a multi-level index for an ISAM sorted file, the student will be able to analyze the trade-off between memory requirements and access times for varying levels and types of index (e.g., dense or sparse).
7. Given the number of frames in a database buffer and a string of logical references (page numbers), the student will be able to produce the corresponding string of physical references (disk blocks) for a variety of page-replacement algorithms, including the optimal, worst, FIFO, LRU and clock algorithms.
8. Given a character set and a list of key strings, the student will be able to construct the corresponding index in the form of either a regular tree or a PATRICIA tree.
9. Given a data file and two (or more) key fields, the student will be able to generate a bitmap index and a 2d-tree index for the file.
10. Given a sequence of record keys and a collision-handling technique (linear probing, quadratic probing, double hashing, linked method with overflow area, or bucket hashing) for static hashing, the student will be able to compute the number of collisions incurred when inserting of the keys into the index.

11. Given an extendible hashing-based index, the student will be able to predict the configuration of its directory file and bucket structure resulting from insertions or deletions of index records.
12. Given the contents of a file, the student will be able to trace the LZ77, LZ78, and LZW algorithms and to produce the corresponding compressed file contents.
13. Given the contents of a file compressed with one of the LZ77, LZ78, and LZW algorithms, the student will be able to decompress the file manually and recover its original contents.
14. Given the description of en encryption scheme such as a substitution cipher or a private-key cryptosystem (e.g., RSA), the student will be able to implement its encryption/decryption algorithm and to simulate it on paper.

Course Grading Policies

Your grade for the course will be based on the following weighted factors:

Factor	**Weight**
Class participation and preparation	10%
Programming Assignments	45%
3 exams:	
Exam 1	15%
Exam 2	15%
Exam 3	15%

At the end of the term, your work in all of these areas will contribute to a numerical grade for the course based on a 100-point scale. Grade cutoff levels on this final scale are:

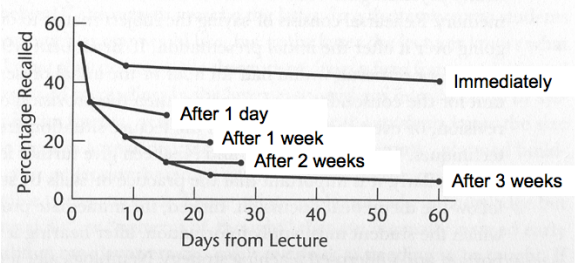
A $\geq$ 92	B $\geq$ 82	C $\geq$ 72	D $\geq$ 62
A- $\geq$ 90	B- $\geq$ 80	C- $\geq$ 70	D- $\geq$ 60
B+ $\geq$ 88	C+ $\geq$ 78	D+ $\geq$ 68	F $<$ 60

FAQ

Do I have to come to class? You are expected to arrive prepared to ALL the course sessions. Furthermore you are expected to participate in the classroom discussions and activities to the best of your abilities. It is difficult to envision a student missing and/or arriving unprepared to a number of the class sessions and still succeeding in the course.

How much time will this course take? Figure about three hours outside of class for each hour in class. That heuristic makes being a full-time student pretty much equivalent to holding a full-time job, so this is really good preparation for the real world that awaits you after graduation.

How can I best prepare for the exams? We've known what the following graph illustrates since 1968:



Bassey M. (1968), in *Learning methods in tertiary education*

Consequently at the end of most of our class periods I will give you one or more exam review problems. The time to work on these review problems is immediately after the material is covered in class. If you have participated in class the day the review problem was distributed, have made a good faith effort to work on the review problem, and are "stuck" on it, I will be more than happy to help you with it if you come my office anytime within three days after you have received the review problem in class.

> Stuckness shouldn't be avoided. It's the psychic predecessor of all real understanding. (Robert Pirsig – Zen and the Art of Motorcycle Maintenance)

After those three days (not counting weekends), *because you have made the choice to not learn effectively*, you are on your own in terms of grappling with these review problems.

What if I'm late in submitting a lab/assignment for evaluation? Each lab and assignment will carry with it a due date. If you are late in submitting it for evaluation, it will be accepted but will be penalized at the rate of 10% of point value the first day late, *an additional* 20% the second, *an additional* 30% the third ...

Is there any way I can carelessly lose points in the course? Yes ...

- Be late in submitting your work for evaluation on labs and assignments.

- Don't participate in and prepare for the class.

What is this class participation/preparation stuff? How does it add up to 10% of my grade? ...

- Be sure to do those review problems before the next class meeting. If you do that and get them right, you get full credit for them. If you do that but get them wrong, you get half credit. If you don't do them, you get no credit.
- Exhibit your knowledge when called on to explain your correct answer to a review problem.
- Exhibit your knowledge when called on to respond to other questions in class

Is there any way I can get some bonus points? ...

- Take advantage of various bonus "challenges" that will accompany some of your programming assignments
- Do an outstanding job when called on to explain your correct answer to a review problem.
- Participate in the ACM Programming Contest on Saturday, October 31 – you will receive an additional $\frac{1}{2}$ point on the 100-point grading scale for each problem that your team of three solves during the contest.

Can I get an extension on work that is due on a specified date? Only if you're ill enough to provide a signed note from the attending physician or have other reasons serious enough that the Dean of Students Office is willing to provide a written note justifying the extension.

If I miss a test, can I make it up? If you are unable to take a scheduled exam, it may be possible to take a make-up exam provided that you do BOTH of the following, which are then subject to my approval:

- Make arrangements prior to the scheduled exam (for last minute emergencies, telephone me at 424-1388 or leave a message at the computer science office, 424-2068). No after-the-fact notifications will be accepted ... *AND*
- Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

Only one make-up exam will be given. It will be a rigorous comprehensive exam given at an arranged time during the last week of the semester.

Can I work with others on assignments? No, not in the sense of two people working on the same program. However, it is acceptable to consult another student for help in debugging a program that you have authored yourself and that is not producing the result you expected. It is also acceptable to cut-and-paste code snippets from in-class demos or examples you find on the internet *provided* that you cite the sources of these code snippets in the introductory documentation block at the beginning of your program.