UNIVERSITY OF WISCONSIN–MADISON
COMPUTER SCIENCES DEPARTMENT

CS 564: DATABASE MANAGEMENT SYSTEMS

Assignment 1: Heap Files
Due: Wednesday, September 27, 1995, 5 p.m.

Instructor: Raghu Ramakrishnan

1 Introduction

Welcome to Minibase, a new version of the Minirel system that has been used in the CS564 course project for many years. Minibase is a small relational DBMS, structured into several layers. In this assignment, you will implement the Heap file layer. You will be given the code for the lower layers (Buffer Manager and Disk Space Manager), as well as the code for managing records on a Heap file page.

You will carry out this assignment, and subsequent ones, in teams of two. Please choose a partner as soon as possible, or send mail to michaell so that we can find a partner for you.

This assignment has three parts. You have to do the following:

1. Familiarize yourself with the Heap file, HFPage, Buffer Manager and Disk Space Manager interfaces.

2. Implement the Heap file class. You can ignore concurrency control and recovery issues, and concentrate on implementing the interface routines. You should deal with free space intelligently, using either a linked list or page directory to identify pages with room for records. When a record is deleted, you must update your information about available free space, and when a record is inserted, you must try to use free space on existing pages before allocating a new page.

3. Run the tests provided by the TA.

1.1 Available Documentation

You should begin by reading the chapter on Disks and Files (available at MACC), to get an overview of the HF layer and buffer management. This material was also covered in class. In addition, HTML documentation is available for Minibase, which you can read using Netscape. There is a link to the Minibase home page in the CS564 home page.

2 Classes to Familiarize Yourself With First

There are four main classes with which you should familiarize yourself: HeapFile, HFPage, BufMgr, DB. A Heap file is seen as a collection of records. Internally, records are stored on a collection of HFPage objects. The code for all these classes will be given to you, with the exception of HeapFile of course.

3 Compiling Your Code and Running the Tests

Copy all the files from */p/course/cs564-1/assigns95/assign1* to your own local assign1 directory and study them carefully. The files are:

- *Makefile:* A sample Makefile for you to compile your project. Study this carefully before you make changes. You will notice that in order to compile your code, you will have to link in the library for the buffer manager, the disk space manager and the record manager */p/course/cs564-1/assigns95/libs/lib1.a*). You need not copy this to your directory. The Makefile automatically compiles your files and links in this library. *You should use the gnu version of 'make', which is in* `/usr/psup/bin`.
- There are several header files, the ones you have to look at are:
  1. *heapfile.h* – heapfile class
  2. *hfpage.h* – hfpage class (a page of the heapfile)
  3. *buf.h* – buffer manager class
  4. *db.h* – the db class
  5. *new_error.h* – the error protocol.

4 What to Turn In, and When

You should turn in copies of your code together with copies of the output produced by running the tests provided by the TAs. The assignment is due at 5PM on September 27th. The solution will be made public after that time, and solutions turned in after that time will not receive any credit. So be sure to turn in whatever you have, even if it is not working fully, at that time.

I emphasize that late submissions will not receive any credit. Computers—and life!—being what they are, expect everything to take longer than you expect, even taking this expectation into account. So start early, and plan on getting things done well before the due date. Nothing short of a nuclear explosion (in the CS building, not the South Pacific) constitutes a valid reason for an extension.