dbdb, the database datebook, or vice−versa.
Project by Aaron Brick, 8/1998−3/2000, under professor Scott Smith.

dbdb is a client−server datebook utility written in C. It is responsive to requests from TCP/IP networks (e.g., the internet) and authenticates its users with passwords. The server is a multithreaded daemon program, meant to be started at boot and stopped at shutdown, along with other daemons on the system. Since the client's functionality is very simple, the included one can be replaced or supplemented by others, designed to run on various machines in various manners.

The primary unit of data is a record, representing an entry in the datebook. The data type contains provisions for the following information:

text (arbitrary−length string)
category (from list of valid categories)
priority (numerical)
dependency (reference to another entry)
start and end times
specificity of start and end times
executable referred to and return value
location (arbitrary−length string)
notes (arbitrary−length string)

These data are maintained at all times by the "session" threads:

entries (array)
categories (??)
sorted order of entries (array of pointers)
preferences
command−line options
entries which depend on another (array of linked lists)

All of a user's records are stored together in a file read and written by gdbm. While a connection with that user is open, the file is open and the data is also stored in memory. The records are stored as a linked list, so that changes made to them run in constant time; there are also several arrays of pointers into the list which exist for the purpose of referring to the list of records in certain orders. Those are regenerated dynamically when necessary. When a change is made to the data it is reflected in both the structures in memory and the file on disk, and the pointer arrays are regenerated.

A global preferences file is kept at `/etc/dbdb.conf`, and a log at `/var/log/dbdb.log`. The server maintains a directory for each user (`/var/dbdb/hostname/username`) with their data file, list of categories, and preferences.

The project has advanced error handling: when it encounters an error, it determines what to print (the error message, possibly including offending values), where to output it (stdout, a log file, the client, or a combination), and whether to quit afterwards. If the client decides to die, it will notify the server thread to also terminate. All connections are closed prior to termination.

The external libraries used are gdbm, an extendible hashing scheme for disk storage of an arbitrary number of records, publib, which provides an important string parsing function, and m, for several mathematical functions.

It was developed under Debian GNU/Linux with the GNU project's development tools: the gcc compiler and gdb debugger.

Since the project is written in C, using POSIX functionality and several widely−available libraries, it should be portable to all UNIX systems. Clients can also be written or implemented on other systems (e.g., desktop or palmtop OSs) which make use of the resources available to the server. A web−based client could be written, to allow access from any browser.

Improvements to be made in the future include:

> The development of more diverse clients, such as ones for palmtop machines, proprietary desktop OSs, or the web. This would allow greater access to the services provided by one UNIX server.
>
> Encrypted network connections. The password protection functionality is really moot until the programs use secure connections.
>
> Various output formats, such as HTML. This could be written into each client; a web−based one would receive lines output from the server and parse them into an attractive HTML table.