【Linux Programming】Day23

Tags	
Date	@June 21, 2022
Summary	gdbm Access Functions

【Ch7】Data Management

7.2 Database

The `dbm` database enables us store data structures of variable size, using an index, and then retrieve the structure either using the index or by sequentially scanning the database.

The `dbm` database's basic element is a block of data to store, coupled with a companion block of data that acts as a key for retrieving that data.

To manipulate the data blocks, the `ndbm.h` include file defines a new type called `datum`.

The `datum` structure must include the following members.

```
struct datum {
  void *dptr;
  size_t dsize;
};
```

Note: My note uses gdbm instead of dbm. The interfaces are slightly different.

7.2.1 dbm Access Functions

```
#include <gdbm.h>

GDBM_FILE *dbm_open(const char *filename, int block_size, int file_open_flags, mode_t file_mode, NULL);
int gdbm_store(DBM *database_descriptor, datum key, datum content, int store_mdoe);
datum gdbm_fetch(DBM *database_descriptor, datum key);
void gdmb_close(DBM *database_descriptor);
```

gdbm_open

This function is used to open existing databases and can be used to create new databases. The `filename` argument is a base filename, without a `.dir` or `.pag` extension.

The `block_size` parameter is used only when `gdbm_open` has to create a new database file. It represents the size of a single transfer from disk to memory.

gdbm_store

We use this function to enter data into the database. All data must be stored with a unique index.

To define the data that we wish to store and the index used to refer to it, we must set up two datum types: one to refer to the index and one for the actual data.

The `store_mode` argument defines what happens if an attempt is made to store some data using a key that already exists.

If `GDBM_INSERT` is set, the store fails and `gdbm_store` returns 1.

If it is set to `GDBM_REPLACE`, the new data overwrites the existing data and `gdbm_store` returns 0.

When an error occurs, `gdbm_store` returns -1.

gdbm_fetch

The `gdbm_fetch` routine is used for retrieving data from the database. It takes a `gdbm` pointer and a `datum` type, which must be set up to point to a key.

If the required value is not found, the `dptr` will be set to `Null`.