CS 2250 | **Data Structures and Problem Solving**

Assignment # 3 | Due Date: March 22, 2007

Files

Create a directory ${USER}.3 in your home. Keep all programs and datafiles for this assignment in this directory. Do each of the assigned programs in a separate directory within this directory. After you are done with the assignment, remove the executables, and execute the following commands on admiral:

```
% cd
% ~sanjiv/bin/handin cs2250 3
```

1. Write a transaction processing system to update accounts in a bank. You need to handle five files, two of which are binary files. The files are

   master This file is a binary file and contains information about existing accounts in the bank. The format of the file is:

Field name	Field type	Field width
acct_num	integer	7
last_name	string	20
first_name	string	20
balance	float	10, including two decimals
last_trans_date	date	packed integer structure

   The structure type date is made up of bit fields as

month	4-bit integer
day	5-bit integer
year	15-bit integer

   transaction This is an ASCII file and keeps transactions for the day. The format of the file is:

Field name	Field type	Field width
acct_num	integer	7
trans_amt	float	10, including two decimals
trans_type	character	1
trans_completed	character	1
new_acct	character	1

   I want you to use bit fields to encode the last two field names. The field trans_type can have the value c (for credit) or d (for debit). Transaction completed and new account fields can have a value true or false.

   new_master This is also a binary file and is a result of updating the existing master file with the transaction file. This file has the same format as the master file.

new_acct File containing information about new accounts, in ASCII format.

error_log Output file (in ASCII) to report any errors.

Write a menu driven program that will allow a teller to give a certain amount of money to a customer, based on the customer number. Of course, you must check that the customer has enough balance in his/her account. However, do not update the master file at this point but simply append the transaction to the transaction file. Notice that this will allow a person to draw more money if he/she performs more than one withdrawal transaction. If any money is deposited, it is also recorded in the transaction file without any updates performed on the master file. The master file is always updated at the end of the day.

To update the master file, you should read in the transaction file and sort its entries in memory using the Unix library function `qsort`. The entries are sorted based on `acct_num` and `trans_type` such that the credit transactions for the account are performed before the debit transactions. The description of the `qsort` function, as well as an example, is in the man pages. The master files are always kept in the sorted order. You will read one entry from the master file, and see if there is a completed transaction against it. If the transaction contains a new account (one that does not exist in the master file), you have to get the corresponding information from the `new_acct` file.

Finally, any uncompleted transactions, new accounts, and errors, are recorded in the file `error_log`, along with the complete transaction that failed. Also add the reason for the error log entry (one of the three mentioned above). An error is also generated if the corresponding account does not exist in the master file.