Bin2text.c explanation:

1. Create a struct to hold variables to be read.
2. Check for argc and argv correct size.
3. Open in file & out file + check for errors.
4. Read into struct from binary file exactly the size of struct.
5. Write in all variables of the struct, with correct format.

Text2bin.c explanation:

1. Create a struct to hold variables to be read.
2. Check for argc and argv correct size.
3. Open in file & out file + check for errors.
4. Parse each line, read in correct sizes into struct.
5. Write struct into binary file and repeat.

Bin2indexed.c explanation:

Comments in my code explain each step, however I'll explain the general process here.

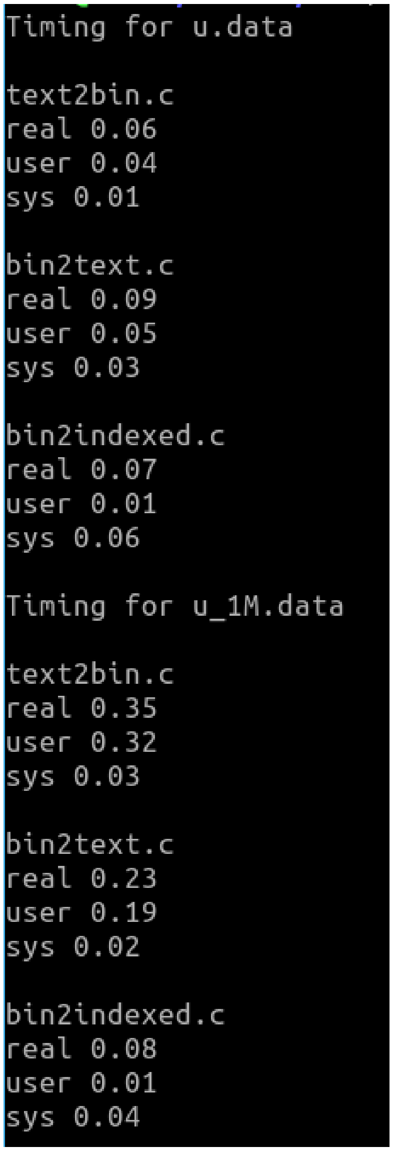
1. I open the item file, and initialize an empty long int pointer to hold offsets.
2. Read each line, get the movie_index and reallocate space in offsets to hold up to movie_index long ints.
3. Get the current offset using ftell(), and put it into offsets at movie_index * size_of(int long).

Now we have a pointer which holds all the offsets. We'll open the binary file and a new output file, read in the data in the binary file to the size of struct (same as bin2text).

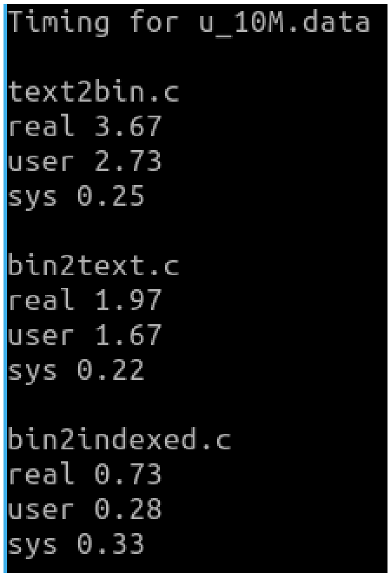
1. Copy the values of struct into struct2, which is the same as the struct in bin2text except it's item_id is of size long_int to accommodate the offsets.
2. Change the item_id of struct2 to the corresponding offset.
3. Write struct2 to the output file.
4. Close files and free data.

Timings

Notice that as the file sizes increase by powers of ten, so do the process execution times. That is, our programs scale linearly.



```
Timing for u.data

text2bin.c
real 0.06
user 0.04
sys 0.01

bin2text.c
real 0.09
user 0.05
sys 0.03

bin2indexed.c
real 0.07
user 0.01
sys 0.06

Timing for u_1M.data

text2bin.c
real 0.35
user 0.32
sys 0.03

bin2text.c
real 0.23
user 0.19
sys 0.02

bin2indexed.c
real 0.08
user 0.01
sys 0.04
```



```
Timing for u_10M.data

text2bin.c
real 3.67
user 2.73
sys 0.25

bin2text.c
real 1.97
user 1.67
sys 0.22

bin2indexed.c
real 0.73
user 0.28
sys 0.33
```