Timing and Approaches

Daniel Kramer

text2bin.c

Approach: I read all text file lines with fgets, strtok that result into four parts, atoi each part and cast accordingly, then fwrite the address for each part in order into the binary file

Timing results:

	100K	1M
real	0.42	3.72
user	0.06	0.52
system	0.00	0.03

bin2text.c

Approach: I fread all four original parts individually by specifying their sizes, and fprintf into the text file separated by the appropriate characters.

Timing Results:

	100K	1M
real	0.56	6.63
user	0.07	0.61
system	0.00	0.05

bin2indexed.c

Approach: I malloc a long array and read offsets into there by fgetting lines and then ftell-ing to find the offset of the next item. If the long array fills up, I realloc it twice its original size. I then fread four parts like in bin2text, except for the second short, I fwrite the value at its position-1 in the offsets array.

Timing Results:

	100K	1M
real	0.42	5.11
user	0.02	0.18
system	0.00	0.04

Scaling:

- All three programs seemed to scale relatively linearly to the amount of rows of initial data. I.e. for a 10x increase in reviews, the user and real times increased ~tenfold while the system times were too small to track.
- On Linux Lab, programs generally took about 25-50x longer to run than on my local machine most likely influenced by connection quality and difference in resources.