CSC369 Assignment 3 Write-Up

Name: Arvind Ramesh (vangalna), Temisan Iwere (iweretem)

Algorithm 1: FIFO

Program 1: tr-simpleloop.ref

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Hit rate	72.0162	74.0117	74.3882	74.4635
Hit count	7651	7863	7903	7911
Miss count	2973	2761	2721	2713
Overall eviction count	2923	2661	2571	2513
Clean eviction count	126	32	8	6
Dirty eviction count	2797	2629	2563	2507

Program 2: tr-matmul.ref

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Hit rate	96.2873	96.5338	99.3842	99.4063
Hit count	174175	174621	179777	179817
Miss count	6716	6270	1114	1074
Overall eviction count	6666	6170	964	874
Clean eviction count	2860	2693	171	88
Dirty eviction count	3806	3477	793	786

CONTINUED ON NEXT PAGE….

Program 3: tr-blocked.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	99.7020	99.7891	99.7957	99.8291
Hit count	953517	954350	954413	954733
Miss count	2850	2017	1954	1634
Overall eviction count	2800	1917	1804	1434
Clean eviction count	826	617	584	438
Dirty eviction count	1974	1300	1220	996

Program 4: tr-heaploop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	98.4247	98.4577	98.4577	98.6722
Hit count	11934	11938	11938	11964
Miss count	191	187	187	161
Overall eviction count	141	87	37	0
Clean eviction count	2	0	0	0
Dirty eviction count	139	87	37	0

Algorithm 2: exact LRU

Program 1: tr-simpleloop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	73.8046	74.6800	74.7082	74.7082
Hit count	7841	7934	7937	7937
Miss count	2783	2690	2687	2687
Overall eviction count	2733	2590	2537	2487
Clean eviction count	67	2	0	0
Dirty eviction count	2666	2588	2537	2487

Program 2: tr-matmul.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	96.6344	96.7699	99.4206	99.4273
Hit count	174803	175048	179843	179855
Miss count	6088	5843	1048	1036
Overall eviction count	6038	5743	898	836
Clean eviction count	2679	2549	129	78
Dirty eviction count	3359	3194	769	758

Program 3: tr-blocked.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	99.7588	99.8118	99.8125	99.8148
Hit count	954060	954567	954574	954596
Miss count	2307	1800	1793	1771
Overall eviction count	2257	1700	1643	1571
Clean eviction count	643	585	565	534
Dirty eviction count	1614	1115	1078	1037

Program 4: tr-heaploop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	98.4742	98.4742	98.5567	98.6722
Hit count	11940	11940	11950	11964
Miss count	185	185	175	161
Overall eviction count	135	85	25	0
Clean eviction count	0	0	0	0
Dirty eviction count	135	85	25	0

Algorithm 3: CLOCK

Program 1: tr-simpleloop.ref

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Hit rate	73.6446	74.6611	74.6894	74.6988
Hit count	7824	7932	7935	7936
Miss count	2800	2692	2689	2688
Overall eviction count	2750	2592	2539	2488
Clean eviction count	74	2	0	0
Dirty eviction count	2676	2590	2539	2488

Program 2: tr-matmul.ref

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Hit rate	96.6250	96.7085	99.4151	99.4267
Hit count	174786	174937	179833	179854
Miss count	6105	5954	1058	1037
Overall eviction count	6055	5854	908	837
Clean eviction count	2686	2604	129	78
Dirty eviction count	3369	3250	779	759

Program 3: tr-blocked.ref

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Hit rate	99.7563	99.8036	99.8087	99.8284
Hit count	954036	954489	954537	954726
Miss count	2331	1878	1830	1641
Overall eviction count	2281	1778	1680	1441
Clean eviction count	656	585	564	470
Dirty eviction count	1625	1193	1116	971

Program 4: tr-heaploop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	98.4577	98.4660	98.4660	98.6722
Hit count	11938	11939	11939	11964
Miss count	187	186	186	161
Overall eviction count	137	86	36	0
Clean eviction count	0	0	0	0
Dirty eviction count	137	86	36	0

Algorithm 4: OPT

Program 1: tr-simpleloop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	74.8023	75.1506	75.1506	75.1506
Hit count	7947	7984	7984	7984
Miss count	2677	2640	2640	2640
Overall eviction count	2627	2540	2490	2440
Clean eviction count	24	0	0	0
Dirty eviction count	2603	2540	2490	2440

Program 2: tr-matmul.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	97.9325	99.3068	99.4975	99.5251
Hit count	177151	179637	179982	180032
Miss count	3740	1254	909	859
Overall eviction count	3690	1154	759	659
Clean eviction count	1499	225	38	2
Dirty eviction count	2191	929	721	657

Program 3: tr-blocked.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	99.8159	99.8432	99.8594	99.8698
Hit count	954606	954867	955022	955122
Miss count	1761	1500	1345	1245
Overall eviction count	1711	1400	1195	1045
Clean eviction count	578	438	335	270
Dirty eviction count	1133	962	860	775

Program 4: tr-heaploop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	98.5155	98.5155	98.5155	98.6722
Hit count	11945	11945	11945	11964
Miss count	180	180	180	161
Overall eviction count	130	80	30	0
Clean eviction count	9	9	9	0
Dirty eviction count	121	71	21	0

Algorithm 5: RAND

Program 1: tr-simpleloop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	71.6867	73.8611	74.3788	74.3976
Hit count	7616	7847	7902	7904
Miss count	3008	2777	2722	2720
Overall eviction count	2958	2677	2572	2520
Clean eviction count	151	42	13	12
Dirty eviction count	2807	2635	2559	2508

Program 2: tr-matmul.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	96.6123	98.5947	99.2161	99.3327
Hit count	174763	178349	179473	179684
Miss count	6128	2542	1418	1207
Overall eviction count	6078	2442	1268	1007
Clean eviction count	2606	881	316	186
Dirty eviction count	3472	1561	952	821

Program 3: tr-blocked.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	99.6355	99.7586	99.7884	99.8076
Hit count	952881	954058	954343	954527
Miss count	3486	2309	2024	1840
Overall eviction count	3436	2209	1874	1640
Clean eviction count	1170	717	601	514
Dirty eviction count	2266	1492	1273	1126

Program 4: tr-heaploop.ref

	Mem. size 50	Mem. size 100	Mem. size 150	Mem. size 200
Hit rate	98.4577	98.6062	98.6722	98.6722
Hit count	11938	11956	11964	11964
Miss count	187	169	161	161
Overall eviction count	137	69	11	0
Clean eviction count	7	1	0	0
Dirty eviction count	130	68	11	0

*Note: Interesting program was **heaploop.c**, a program from Exercise 8. **tr-heaploop.ref** was obtained from **heaploop.c**.*

Comparison of the algorithms

One common pattern noticed with all of the algorithms and the trace files is that the hit count as well as the corresponding hit rate are directly proportional to the memory size. In other words, as the size of memory increases, the hit count and the corresponding hit rate increases.
On the contrast, the miss rate and the overall eviction count are inversely proportional to the memory size. They tend to decrease as memory increases. Furthermore, for the trace file ***tr-heaploop.ref***, we observed that the overall eviction count eventually reaches 0 as memory increases in size, with the clean eviction count usually reaching 0 before the dirty eviction count.

However, we also observed that larger traces have a larger number of memory accesses, resulting in a larger number of hits and a higher hit rate. For example, the trace file ***tr-matmul.ref*** averaged around 180,000 memory references, for all the 5 algorithms. This resulted in the hit rate approaching 100%. The hit rate behaved the same way with the trace file ***tr-blocked.ref***, except this trace averaged around 955,000 memory references.
But we observed that along with an increase in hit rate, the overall eviction count decreases. In this manner, the overall eviction count is inversely proportional to both the size of the memory as well as the hit rate.

All of these observed patterns are similar for all 5 page replacement algorithms, in the sense that the way that each algorithm works doesn't really affect the number of memory references for each of the trace files. In conclusion, all the algorithms gives similar results through different ways of accessing memory and performing actions on it.

Additionally, one interesting observation made is OPT takes the longest time to run each of the trace programs, because of the amount of information it must keep track of to evict pages from memory.

Data for LRU as memory size increases

With the LRU, one common pattern observed is that the hit rate increases as the memory size increases. To demonstrate this, here is some data for ***tr-matmul.ref***:

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Hit rate	96.6344	96.7699	99.4206	99.4273

However, the total eviction count decreases as the memory size increases, no matter the trace file. To demonstrate this, here is some more data for ***tr-matmul.ref***:

	Mem. size 50	**Mem. size 100**	**Mem. size 150**	**Mem. size 200**
Overall eviction count	6038	5743	898	836