

CSCC69 - Page Tables and Replacement Algorithms

Tables from Task 2:

Trace file: tr-blocked.ref							
Algorithm: fifo							
Memsizes	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	99.7318	2411571	6485	6435	4178	2257	
100	99.8211	2413730	4326	4226	2758	1468	
150	99.8255	2413836	4220	4070	2649	1421	
200	99.8689	2414885	3171	2971	1876	1095	
Algorithm: lru							
Memsizes	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	99.7843	2412841	5215	5165	2816	2349	
100	99.8436	2414273	3783	3683	2605	1078	
150	99.8443	2414290	3766	3616	2558	1058	
200	99.8473	2414363	3693	3493	2435	1058	
Algorithm: clock							
Memsizes	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	99.7828	2412805	5251	5201	2859	2342	
100	99.8344	2414052	4004	3904	2618	1286	
150	99.8371	2414117	3939	3789	2574	1215	
200	99.8682	2414870	3186	2986	1927	1059	
Algorithm: opt							
Memsizes	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	99.7674	2412431	5625	5575	2731	2844	
100	99.8184	2413664	4392	4292	1972	2320	
150	99.8509	2414451	3605	3455	1421	2034	
200	99.8651	2414794	3262	3062	1120	1942	
Algorithm: rand							
Memsizes	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	99.6531	2409667	8389	8339	5831	2508	
100	99.7834	2412818	5238	5138	3427	1711	
150	99.8187	2413672	4384	4234	2754	1480	
200	99.8407	2414203	3853	3653	2319	1334	

Trace file: tr-matmul.ref						
Algorithm: ffo						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	60.8711	1757854	1129978	1129928	1084574	45354
100	62.4747	1804164	1083668	1083568	1061237	22331
150	98.8085	2853424	34408	34258	32945	1313
200	98.8266	2853947	33885	33685	32434	1251
Algorithm: lru						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	63.9445	1846609	1041223	1041173	1040068	1105
100	65.1485	1881380	1006452	1006352	1005275	1077
150	98.8613	2854949	32883	32733	31656	1077
200	98.8617	2854960	32872	32672	31595	1077
Algorithm: clock						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	63.9439	1846593	1041239	1041189	1040080	1109
100	63.9517	1846818	1041014	1040914	1039835	1079
150	98.8503	2854631	33201	33051	31971	1080
200	98.8607	2854931	32901	32701	31621	1080
Algorithm: opt						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	66.6792	1983339	904493	904443	592086	312357
100	93.6092	2703275	184557	184457	92131	92326
150	98.1897	2835554	52278	52128	25689	26439
200	98.6994	2850272	37560	37360	18264	19096
Algorithm: rand						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	65.5196	1892096	995736	995686	955990	39696
100	88.7588	2563205	324627	324527	316970	7557
150	96.6702	2791673	96159	96009	93628	2381
200	98.046	2831403	56429	56229	54585	1644

Trace file: tr-simplelinkedlist.ref						
Algorithm: fifo						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	99.6288	79185	295	245	100	145
100	99.815	79333	147	47	0	47
150	99.839	79352	128	0	0	0
200	99.839	79352	128	0	0	0
Algorithm: lru						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	99.7282	79264	216	166	59	107
100	99.8377	79351	129	29	0	29
150	99.839	79352	128	0	0	0
200	99.839	79352	128	0	0	0
Algorithm: clock						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	99.7094	79249	231	181	70	111
100	99.8327	79347	133	33	0	33
150	99.839	79352	128	0	0	0
200	99.839	79352	128	0	0	0
Algorithm: opt						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	99.7735	79300	180	130	22	108
100	99.8188	79336	144	44	1	43
150	99.839	79352	128	0	0	0
200	99.839	79352	128	0	0	0
Algorithm: rand						
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
50	99.6288	79185	295	245	112	133
100	99.815	79333	147	47	2	45
150	99.839	79352	128	0	0	0
200	99.839	79352	128	0	0	0

Trace file: tr-simpleloop.ref							
Algorithm: fifo							
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	70.8399	7220	2972	2922	208	2714	
100	72.9888	7439	2753	2653	45	2608	
150	73.3713	7478	2714	2564	16	2548	
200	73.4498	7488	2708	2508	12	2494	
Algorithm: lru							
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	72.7433	7414	2778	2728	88	2640	
100	73.6852	7510	2682	2582	2	2580	
150	73.7049	7512	2680	2530	0	2530	
200	73.7049	7512	2680	2480	0	2480	
Algorithm: clock							
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	72.6354	7403	2789	2739	97	2642	
100	73.6754	7509	2683	2583	2	2581	
150	73.6852	7510	2682	2532	0	2532	
200	73.6951	7511	2681	2481	0	2481	
Algorithm: opt							
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	73.3909	7480	2712	2662	28	2634	
100	73.9011	7532	2660	2560	2	2558	
150	70.0157	7136	3056	2906	0	2906	
200	70.0942	7144	3048	2848	0	2848	
Algorithm: rand							
Memsize	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count	
50	70.9282	7229	2963	2913	219	2694	
100	73.0181	7442	2750	2650	50	2600	
150	73.3222	7473	2719	2569	18	2551	
200	73.4792	7489	2703	2503	14	2489	

Algorithm Comparisons

In general the opt algorithm performs the best as expected. Since it has access to the actual trace files, it can optimally compute which frame is the best to evict based on when the frame is next used.

We also noticed that clock performs the second best, with similar hit rates to opt. The rand algorithm performs consistently and is hence a good lower bound for assessing performance. We do not observe Belady's anomaly while running fifo in all cases. This is because the allocation algorithm runs just ahead of fifo by allocating pages that fifo immediately evicts. In other words, the order of allocation is the best case for fifo and the increase in memsize does not affect this as the fifo algorithm resets the counter exactly the same time as the allocation resets to 0.

LRU Description

As shown in the tables above, the data shows that LRU typically performs poorly when there is little memory. But as memory is increased, the hit rate also increases which supports how the LRU algorithm works. Since LRU keeps track of the least used frames, the more memory means its more accurately able to store information about which frame has been the least used and correctly evicts that page, which would result in a higher hit rate.