Looking at the results table, we can deduce that most of the algorithms had similar results. For the case of tr-simpleloop. Most algorithms had similar results on each of the different traces when compared to each other. For the case of tr-simpleloop, the hit rate % of each of the algorithms were around ~75%, as oppose to the ~90+% hit rates from the other algorithms. The case for this might be since simpleloop goes through new elements, without accessing them again which leads to a lower hit rate. FIFO and CLOCK have very similar results according to the table. OPT as expected has the highest hit rate compared to all the other algorithms as it is allowed the benefit to “see into the future”, thus allowing the algorithm to remove frames which will cause the least amount of miss rates.

LRU Performance

My LRU page replacement algorithm was implemented using a single linked list data structure to store the least recently used page at the head of the list and the recently used page at the end.
The hit rate performance for this algorithm when tested using the trace files did not suffer considerably in performance when compared to other algorithms, however it should be noted that it did not beat or do exceptionally well when contrasted with the other algorithms’ performance. In fact, in tr-matmul the hit-rate performance was horrible using a small memory cache space. It should be noted from the table that as memory increases, so does the hit rate for LRU. This is due to the fact that as the cache space for our memory increases, the pages we can store increases and it will lead to more recent pages in our list, which will increase our hit rate. The chances of it hitting a page in our list that was not recently used, would decrease as well. LRU with a decently sized memory cache performs well for programs that loop through a certain array of elements which does various instructions on those elements. It should be noted that the overhead cost for LRU is high, as oppose to FIFO, rand or CLOCK algorithms. For every reference, we need to move the frame in our list to move to the end of our list, to ensure least recently used. This is costly in performance cost where we need to update our list, iterating through O(n) elements, where n is the frames in physmem. As a result, the algorithm will run considerably slower then FIFO, rand or CLOCK.

CSC369a2 – Result Table

ALGO	MEMORY SIZE	TRACE FILE	Hit Rate:	Hit Count:	Miss Count:	Overall evictions:	Clean evictions:	Dirty evictions:
Rand	50	tr-simpleloop	72.6	8015	3025	2975	274	2701
Rand	100	tr-simpleloop	75.045	8285	2755	2655	46	2609
Rand	150	tr-simpleloop	75.426	8327	2713	2563	17	2546
Rand	200	tr-simpleloop	75.417	8326	2714	2514	16	2498
FIFO	50	tr-simpleloop	72.944	8053	2987	2937	212	2725
FIFO	100	tr-simpleloop	75.018	8282	2758	2658	45	2613
FIFO	150	tr-simpleloop	75.389	8323	2717	2567	16	2551
FIFO	200	tr-simpleloop	75.462	8331	2709	2509	12	2497
LRU	50	tr-simpleloop	74.719	8249	2791	2741	97	2644
LRU	100	tr-simpleloop	75.688	8356	2684	2584	0	2584
LRU	150	tr-simpleloop	75.697	8357	2683	2533	0	2533
LRU	200	tr-simpleloop	75.697	8357	2683	2483	0	2483
CLOCK	50	tr-simpleloop	74.647	8241	2799	2749	104	2645
CLOCK	100	tr-simpleloop	75.661	8353	2687	2587	2	2585
CLOCK	150	tr-simpleloop	75.688	8356	2684	2534	0	2534
CLOCK	200	tr-simpleloop	75.688	8356	2684	2484	0	2484
OPT	50	tr-simpleloop	75.797	8368	2672	2622	24	2598
OPT	100	tr-simpleloop	76.096	8401	2639	2539	0	2539
OPT	150	tr-simpleloop	76.096	8401	2639	2489	0	2489
OPT	200	tr-simpleloop	76.096	8401	2639	2439	0	2439

CSC369a2 – Result Table

ALGO	MEMORY SIZE	TRACE FILE	Hit Rate:	Hit Count:	Miss Count:	Overall evictions:	Clean evictions:	Dirty evictions:
Rand	50	tr-matmul	66.48	1973240	994912	994862	955503	39359
Rand	100	tr-matmul	89.108	2644847	323305	323205	315719	7486
Rand	150	tr-matmul	96.769	2872249	95903	95753	93431	2322
Rand	200	tr-matmul	98.107	2911955	56197	55997	54358	1639
FIFO	50	tr-matmul	61.93	1838164	1129988	1129938	1084592	45346
FIFO	100	tr-matmul	63.49	1884472	1083680	1083580	1061242	22338
FIFO	150	tr-matmul	98.841	2933746	34406	34256	32947	1309
FIFO	200	tr-matmul	98.858	2934270	33882	33682	32435	1247
LRU	50	tr-matmul	64.92	1926933	1041219	1041169	1040071	1098
LRU	100	tr-matmul	66.092	1961708	1006444	1006344	1005269	1075
LRU	150	tr-matmul	98.892	2935275	32877	32727	31654	1073
LRU	200	tr-matmul	98.893	2935286	32866	32666	31593	1073
CLOCK	50	tr-matmul	64.92	1926923	1041229	1041179	1040080	1099
CLOCK	100	tr-matmul	66.249	1966370	1001782	1001682	1000607	1075
CLOCK	150	tr-matmul	98.833	2933499	34653	34503	33428	1075
CLOCK	200	tr-matmul	98.892	2935273	32879	32679	31606	1073
OPT	50	tr-matmul	80.211	2381183	587457	587407	586321	1086
OPT	100	tr-matmul	96.874	2875843	92797	92697	91613	1084
OPT	150	tr-matmul	99.103	2942026	26614	26464	25380	1084
OPT	200	tr-matmul	99.351	2949376	19264	19064	17980	1084
ALGO	MEMORY SIZE	TRACE FILE	Hit Rate:	Hit Count:	Miss Count:	Overall evictions:	Clean evictions:	Dirty evictions:

CSC369a2 – Result Table

Rand	50	tr-blocked	99.672	2516731	8293	8243	5726	2517
Rand	100	tr-blocked	99.793	2519796	5228	5128	3378	1750
Rand	150	tr-blocked	99.824	2520591	4433	4283	2802	1481
Rand	200	tr-blocked	99.849	2521210	3814	3614	2285	1329
FIFO	50	tr-blocked	99.744	2518567	6457	6407	4158	2249
FIFO	100	tr-blocked	99.829	2520705	4319	4219	2745	1474
FIFO	150	tr-blocked	99.832	2520787	4237	4087	2668	1419
FIFO	200	tr-blocked	99.874	2521851	3173	2973	1881	1092
LRU	50	tr-blocked	99.795	2519846	5178	5128	2786	2342
LRU	100	tr-blocked	99.849	2521215	3809	3709	2631	1078
LRU	150	tr-blocked	99.849	2521223	3801	3651	2595	1056
LRU	200	tr-blocked	99.854	2521331	3693	3493	2437	1056
CLOCK	50	tr-blocked	99.772	2519259	5765	5715	3304	2411
CLOCK	100	tr-blocked	99.835	2520852	4172	4072	2638	1434
CLOCK	150	tr-blocked	99.849	2521218	3806	3656	2599	1057
CLOCK	200	tr-blocked	99.872	2521803	3221	3021	1953	1068
OPT	50	tr-blocked	99.852	2521802	3742	3692	2606	1086
OPT	100	tr-blocked	99.88	2522509	3035	2935	1860	1075
OPT	150	tr-blocked	99.9	2523006	2538	2388	1313	1075
OPT	200	tr-blocked	99.909	2523256	2288	2088	1024	1064
ALGO	MEMORY SIZE	TRACE FILE	Hit Rate:	Hit Count:	Miss Count:	Overall evictions:	Clean evictions:	Dirty evictions:

CSC369a2 – Result Table

Rand	50	tr-malloc_sys.ref	99.965	1736660	606	556	311	245
Rand	100	tr-malloc_sys.ref	99.985	1737001	265	165	27	138
Rand	150	tr-malloc_sys.ref	99.988	1737056	210	60	2	58
Rand	200	tr-malloc_sys.ref	99.989	1737069	197	0	0	0
FIFO	50	tr-malloc_sys.ref	99.969	1736736	530	480	244	236
FIFO	100	tr-malloc_sys.ref	99.986	1737026	240	140	17	123
FIFO	150	tr-malloc_sys.ref	99.988	1737061	205	55	0	55
FIFO	200	tr-malloc_sys.ref	99.989	1737069	197	0	0	0
LRU	50	tr-malloc_sys.ref	99.98	1736917	349	299	123	176
LRU	100	tr-malloc_sys.ref	99.988	1737061	205	105	2	103
LRU	150	tr-malloc_sys.ref	99.989	1737068	198	48	0	48
LRU	200	tr-malloc_sys.ref	99.989	1737069	197	0	0	0
CLOCK	50	tr-malloc_sys.ref	99.98	1736921	345	295	116	179
CLOCK	100	tr-malloc_sys.ref	99.988	1737053	213	113	8	105
CLOCK	150	tr-malloc_sys.ref	99.988	1737066	200	50	0	50
CLOCK	200	tr-malloc_sys.ref	99.989	1737069	197	0	0	0
OPT	50	tr-malloc_sys.ref	99.987	1737033	233	183	28	155
OPT	100	tr-malloc_sys.ref	99.989	1737069	197	97	0	97
OPT	150	tr-malloc_sys.ref	99.989	1737069	197	47	0	47
OPT	200	tr-malloc_sys.ref	99.989	1737069	197	0	0	0