CSC 369 – Operating Systems

Assignment 2 Writeup

Julianna Paprakis

g4paprak

996010472

Contents

Tables ....

Size 50 1

Size 100 2

Size 150 3

Size 200 4

Writeup ....

Comparison Paragraph 5

LRU Description.. 5

Tables

Size 50

Running with Size = 50								
Program	Eviction Algo	Hit Rate	Hit count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count	Difference from Opt
simpleloop	FIFO	73.1318%	8,103	2,977	**2,927**	207	2,720	-2.7527%
simpleloop	LRU	74.8105%	8,289	2,791	**2,741**	97	2,644	-1.0740%
simpleloop	CLOCK	74.6841%	8,275	2,805	**2,755**	109	2,646	-1.2004%
simpleloop	RAND	72.6986%	8,055	3,025	**2,975**	274	2,701	-3.1859%
simpleloop	OPT	75.8845%	8,408	2,672	**2,622**	24	2,598	0.0000%
matmul	FIFO	62.0302%	1,841,608	1,127,280	**1,127,230**	1,083,235	43,995	-18.1827%
matmul	LRU	64.9285%	1,927,655	1,041,233	**1,041,183**	1,040,078	1,105	-15.2844%
matmul	CLOCK	64.9277%	1,927,632	1,041,256	**1,041,206**	1,040,096	1,110	-15.2852%
matmul	RAND	66.4804%	1,973,728	995,160	**995,110**	955,918	39,192	-13.7325%
matmul	OPT	80.2129%	2,381,431	587,457	**587,407**	586,321	1,086	0.0000%
blocked	FIFO	99.7437%	2,519,286	6,474	**6,424**	4,165	2,259	-0.1082%
blocked	LRU	99.7938%	2,520,553	5,207	**5,157**	2,808	2,349	-0.0580%
blocked	CLOCK	99.7928%	2,520,527	5,233	**5,183**	2,853	2,330	-0.0590%
blocked	RAND	99.6675%	2,517,362	8,398	**8,348**	5,808	2,540	-0.1843%
blocked	OPT	99.8518%	2,522,018	3,742	**3,692**	2,606	1,086	0.0000%
do_fstree	FIFO	98.2466%	309,068	5,516	**5,466**	4,527	939	-1.5080%
do_fstree	LRU	99.1595%	311,940	2,644	**2,594**	2,265	329	-0.5951%
do_fstree	CLOCK	99.1309%	311,850	2,734	**2,684**	2,241	443	-0.6237%
do_fstree	RAND	98.8458%	310,953	3,631	**3,581**	2,881	700	-0.9088%
do_fstree	OPT	99.7546%	313,812	772	**722**	423	299	0.0000%

Size 100

Running with Size = 100								
Program	Eviction Algo	Hit Rate	Hit count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count	Difference from Opt
simpleloop	FIFO	75.1083%	8,322	2,758	**2,658**	45	2,613	-1.0740%
simpleloop	LRU	75.7762%	8,396	2,684	**2,584**	-	2,584	-0.4061%
simpleloop	CLOCK	75.7491%	8,393	2,687	**2,587**	2	2,585	-0.4332%
simpleloop	RAND	75.1354%	8,325	2,755	**2,655**	46	2,609	-1.0469%
simpleloop	OPT	76.1823%	8,441	2,639	**2,539**	-	2,539	0.0000%
matmul	FIFO	63.5031%	1,885,336	1,083,552	**1,083,452**	1,061,226	22,226	-33.3712%
matmul	LRU	66.1000%	1,962,436	1,006,452	**1,006,352**	1,005,271	1,081	-30.7743%
matmul	CLOCK	64.9345%	1,927,834	1,041,054	**1,040,954**	1,039,871	1,083	-31.9398%
matmul	RAND	89.1101%	2,645,579	323,309	**323,209**	315,794	7,415	-7.7643%
matmul	OPT	96.8744%	2,876,091	92,797	**92,697**	91,613	1,084	0.0000%
blocked	FIFO	99.8273%	2,521,397	4,363	**4,263**	2,780	1,483	-0.0526%
blocked	LRU	99.8483%	2,521,928	3,832	**3,732**	2,647	1,085	-0.0316%
blocked	CLOCK	99.8392%	2,521,699	4,061	**3,961**	2,648	1,313	-0.0406%
blocked	RAND	99.7943%	2,520,564	5,196	**5,096**	3,390	1,706	-0.0856%
blocked	OPT	99.8798%	2,522,725	3,035	**2,935**	1,860	1,075	0.0000%
do_fstree	FIFO	99.8935%	314,249	335	**235**	47	188	-0.0423%
do_fstree	LRU	99.9237%	314,344	240	**140**	1	139	-0.0121%
do_fstree	CLOCK	99.9218%	314,338	246	**146**	5	141	-0.0140%
do_fstree	RAND	99.8910%	314,241	343	**243**	60	183	-0.0448%
do_fstree	OPT	99.9358%	314,382	202	**102**	-	102	0.0000%

Size 150

Running with Size = 150								
Program	Eviction Algo	Hit Rate	Hit count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count	Difference from Opt
simpleloop	FIFO	75.4783%	8,363	2,717	**2,567**	16	2,551	-0.7040%
simpleloop	LRU	75.7852%	8,397	2,683	**2,533**	-	2,533	-0.3971%
simpleloop	CLOCK	75.7581%	8,394	2,686	**2,536**	-	2,536	-0.4242%
simpleloop	RAND	75.5144%	8,367	2,713	**2,563**	17	2,546	-0.6679%
simpleloop	OPT	76.1823%	8,441	2,639	**2,489**	-	2,489	0.0000%
matmul	FIFO	98.8409%	2,934,476	34,412	**34,262**	32,946	1,316	-0.2627%
matmul	LRU	98.8923%	2,936,003	32,885	**32,735**	31,655	1,080	-0.2112%
matmul	CLOCK	98.8815%	2,935,681	33,207	**33,057**	31,974	1,083	-0.2221%
matmul	RAND	96.7433%	2,872,201	96,687	**96,537**	94,179	2,358	-2.3602%
matmul	OPT	99.1036%	2,942,274	26,614	**26,464**	25,380	1,084	0.0000%
blocked	FIFO	99.8312%	2,521,496	4,264	**4,114**	2,686	1,428	-0.0683%
blocked	LRU	99.8486%	2,521,935	3,825	**3,675**	2,611	1,064	-0.0510%
blocked	CLOCK	99.8430%	2,521,795	3,965	**3,815**	2,615	1,200	-0.0565%
blocked	RAND	99.8248%	2,521,335	4,425	**4,275**	2,778	1,497	-0.0747%
blocked	OPT	99.8995%	2,523,222	2,538	**2,388**	1,313	1,075	0.0000%
do_fstree	FIFO	99.9263%	314,352	232	**82**	-	82	-0.0095%
do_fstree	LRU	99.9313%	314,368	216	**66**	-	66	-0.0045%
do_fstree	CLOCK	99.9288%	314,360	224	**74**	-	74	-0.0070%
do_fstree	RAND	99.9205%	314,334	250	**100**	8	92	-0.0153%
do_fstree	OPT	99.9358%	314,382	202	**52**	-	52	0.0000%

Size 200

Running with Size = 200								
Program	Eviction Algo	Hit Rate	Hit count	Miss count	Overall eviction count	Clean eviction count	Dirty eviction count	Difference from Opt
simpleloop	FIFO	75.5505%	8,371	2,709	**2,509**	12	2,497	-0.6318%
simpleloop	LRU	75.7852%	8,397	2,683	**2,483**	-	2,483	-0.3971%
simpleloop	CLOCK	75.7762%	8,396	2,684	**2,484**	-	2,484	-0.4061%
simpleloop	RAND	75.5054%	8,366	2,714	**2,514**	16	2,498	-0.6769%
simpleloop	OPT	76.1823%	8,441	2,639	**2,439**	-	2,439	0.0000%
matmul	FIFO	98.8585%	2,934,998	33,890	**33,690**	32,435	1,255	-0.4926%
matmul	LRU	98.8927%	2,936,014	32,874	**32,674**	31,594	1,080	-0.4584%
matmul	CLOCK	98.8918%	2,935,986	32,902	**32,702**	31,620	1,082	-0.4594%
matmul	RAND	98.1003%	2,912,487	56,401	**56,201**	54,572	1,629	-1.2509%
matmul	OPT	99.3511%	2,949,624	19,264	**19,064**	17,980	1,084	0.0000%
blocked	FIFO	99.8737%	2,522,569	3,191	**2,991**	1,890	1,101	-0.0358%
blocked	LRU	99.8535%	2,522,059	3,701	**3,501**	2,437	1,064	-0.0559%
blocked	CLOCK	99.8719%	2,522,525	3,235	**3,035**	1,969	1,066	-0.0375%
blocked	RAND	99.8472%	2,521,900	3,860	**3,660**	2,326	1,334	-0.0622%
blocked	OPT	99.9094%	2,523,472	2,288	**2,088**	1,024	1,064	0.0000%
do_fstree	FIFO	99.9345%	314,378	206	**6**	-	6	-0.0013%
do_fstree	LRU	99.9358%	314,382	202	**2**	-	2	0.0000%
do_fstree	CLOCK	99.9352%	314,380	204	**4**	-	4	-0.0006%
do_fstree	RAND	99.9358%	314,382	202	**2**	-	2	0.0000%
do_fstree	OPT	99.9358%	314,382	202	**2**	-	2	0.0000%

Writeup

Comparison Paragraph

In the above graphs, I have highlighted the replacement algorithm(s) which has the smallest hit rate difference compared to opt, thus serving as an indicator of the best algorithm for that size and trace. LRU is the best algorithm in 13 out of 16 cases, while clock is usually close to LRU (which maxes sense, as it is just an approximated LRU). Rand ends up performing better for a smaller sizes of physical memory in the matmul trace (50 and 100), and is surprisingly a lot higher in the 100 trace. It may be the case that memory generally is accessed about every 100 accesses for this program, and that LRU is freeing it right around the time that it is accessed again. Since it is random it cannot be relied upon to perform well for all algorithms like matmul. FIFO also surprisingly performs better in the 200 memory blocked trace, but the difference between it and clock is quite small. It is also important to note that after a certain amount of physical memory, opt will not get any better no matter how much memory we add (ie. we will never get a 100% hit rate). This is because we will always have a miss the very first time we are reading memory in.

The trace I uploaded was that of a linked tree which is being created, then garbage collected 50 times. The amount of memory used for this is large, but the amount of unique memory is small (202). As such, all algorithms are able to get over a 99% hit rate on this program!

LRU Description

As can be seen in the table below, as the size of physical memory increases, so do LRU hit rates. It seems that while memory is smaller, the hit rates grow at a much higher rate than when it is bigger, in proportion to the number of unique memory that needs to be accessed. For example, in Matmul, the LRU hit rate grows an entire 33 point change when size is increased from 100 to 150, while the change between 150 and 200 is negligible (less than 0.0004). As well in simpleloop and do_fstree, there is a change of about 1 point and 0.8 points respectively when size grows from 50 to 100, yet all changes after that are miniscule. This is likely because as memory grows, LRU becomes more accurate as it has more options to choose from, and those memory which are used frequently are almost guaranteed to remain in physical memory. In the do_fstree of size 200, LRU ends up being the same as opt (although this is mostly due to the fact that do_fstree only has about 202 unique memory accesses).

LRU hit rates as memory size increases

Trace	Size 50	Size 100	Size 150	Size 200
Simpleloop	74.8105%	75.7762%	75.7852%	75.7852%
Matmul	64.9285%	66.1000%	98.8923%	98.8927%
Blocked	99.7938%	99.8483%	99.8486%	99.8535%
do_fstree	99.1595%	99.9237%	99.9313%	99.9358%