simpleloop.c

MEMSIZE = 50, simpleloop.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	53	3	2	1
CLOCK	-nan	0	53	3	2	1
FIFO	-nan	0	53	3	3	0
LRU						
OPT						

MEMSIZE = 100, simpleloop.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	53	0	0	0
CLOCK	-nan	0	53	0	0	0
FIFO	-nan	0	53	0	0	0
LRU						
OPT						

MEMSIZE = 150, simpleloop.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	53	0	0	0
CLOCK	-nan	0	53	0	0	0
FIFO	-nan	0	53	0	0	0
LRU						
OPT						

MEMSIZE = 200, simpleloop.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	53	0	0	0
CLOCK	-nan	0	53	0	0	0
FIFO	-nan	0	53	0	0	0
LRU						
OPT						

matmul.c

MEMSIZE = 50, matmul.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count

RAND	-nan	0	61	11	11	0
CLOCK	-nan	0	61	11	11	0
FIFO	-nan	0	61	11	11	0
LRU						
OPT						

MEMSIZE = 100, matmul.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	61	0	0	0
CLOCK	-nan	0	61	0	0	0
FIFO	-nan	0	61	0	0	0
LRU						
OPT						

MEMSIZE = 150, matmul.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	61	0	0	0

CLOCK	-nan	0	61	0	0	0
FIFO	-nan	0	61	0	0	0
LRU						
OPT						

MEMSIZE = 200, matmul.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	61	0	0	0
CLOCK	-nan	0	61	0	0	0
FIFO	-nan	0	61	0	0	0
LRU						
OPT						

blocked.c

MEMSIZE = 50, blocked.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	89	39	39	0
CLOCK	-nan	0	89	39	39	0

FIFO	-nan	0	89	39	39	0
LRU						
OPT						

MEMSIZE = 100, blocked.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	89	0	0	0
CLOCK	-nan	0	89	0	0	0
FIFO	-nan	0	89	0	0	0
LRU						
OPT						

MEMSIZE = 150, blocked.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	89	0	0	0
CLOCK	-nan	0	89	0	0	0
FIFO	-nan	0	89	0	0	0

LRU						
OPT						

MEMSIZE = 200, blocked.c

Algorithm	Hit Rate	Hit Count	Miss Count	Overall Eviction Count	Clean Eviction Count	Dirty Eviction Count
RAND	-nan	0	89	0	0	0
CLOCK	-nan	0	89	0	0	0
FIFO	-nan	0	89	0	0	0
LRU						
OPT						

Comparison of Replacement Algorithms

Explain here why certain algorithms performed better than others.

> I *imagine* they work better because they take into account certain statistics. Statistics that are similar to branch prediction, so they evict the ones that less likely to be used in the future.

How does LRU perform as the memory size increases?

> Uhhh… I *imagine* it does better because there are less evictions!