Pgm4 Description

Based on Figures 8.14 and 8.16

Loop for 1,000 experiments

Generate page address stream (similar to Figure 8.14, see next page)

Loop for working set size from 2 through 20

Faults[working set size][LRU] += LRU( working set size, stream)

Faults[working set size][FIFO] += FIFO ( working set size, stream)

Faults[working set size][Clock] += Clock ( working set size, stream)

Faults[working set size][Random] += Random ( working set size, stream)

End of working set size loop

End of Experiment loop

Loop for working set size from 2 through 20

Test trace with:

Print( working set size,

Faults[working set size][LRU] / 1000.0,

Faults[working set size][FIFO] / 1000.0,

Faults[working set size][Clock] / 1000.0,

Faults[working set size][Random] / 1000.0 )

End of working set size loop

Each function ( LRU, FIFO, Clock, and Random ) count the number of page faults which occur for a page address stream based on the working set size

External to the program, generate a chart that depicts the performance of each of the algorithms (similar to Figure 8.16).

Generate page address stream

Each page address stream is 1,000 page addresses.

for( part=0; part<10; part++ )

Base address equals 25 times a uniform random value in [0,9], inclusive.

for( index=0; index <100; index ++ )

Address[100\*part+index] = Base + normal random number in [0,25], inclusive.

Uniform random number generator

int uniform( int lo, int hi )

{

int x, y = hi – lo + 1, z = RAND\_MAX / y;

while ( y <= (x = ( random() / z )) );

return x + lo;

}

Normal random number generator

int normal ()

{

int x = 0;

for( int i=0; i<5; i++ )

{

x += uniform( 0, 5 );

}

return x;

}