FILE ACCESS

LAB9

WRITE A FILE AND NOTE THE TIME TAKEN

Step 1: Create a random file "dumpfile2" using the following command:

```
vmdhruv@ubuntu:~$ od -A n -t d -N 1000 /dev/urandom > dumpfile2
```

Step 2: Run "dd" command to write the files and see the times taken.

```
vmdhruv@ubuntu:~$ dd if=dumpfile2 of=speetest bs=1M count=100 conv=
fdatasync
0+1 records in
0+1 records out
3063 bytes (3.1 kB, 3.0 KiB) copied, 0.121736 s, 25.2 kB/s
```

Step 3: Repeat the same multiple times and note the time taken. Change the Block size and count size and notice the time. Check if it went through the cache.

```
vmdhruv@ubuntu:~$ dd if=dumpfile2 of=speetest bs=1M count=200 conv=
fdatasync
0+1 records in
0+1 records out
3063 bytes (3.1 kB, 3.0 KiB) copied, 0.0754932 s, 40.6 kB/s
vmdhruv@ubuntu:~$ dd if=dumpfile2 of=speetest bs=1M count=600 conv=
fdatasync
0+1 records in
0+1 records out
3063 bytes (3.1 kB, 3.0 KiB) copied, 0.0158075 s, 194 kB/s
vmdhruv@ubuntu:~$ dd if=dumpfile2 of=speetest bs=1M count=10000 con
v=fdatasync
0+1 records in
0+1 records out
3063 bytes (3.1 kB, 3.0 KiB) copied, 0.0960681 s, 31.9 kB/s
vmdhruv@ubuntu:~$ dd if=dumpfile2 of=speetest bs=1M count=20000 con
v=fdatasync
0+1 records in
0+1 records out
3063 bytes (3.1 kB, 3.0 KiB) copied, 0.0429687 s, 71.3 kB/s
vmdhruv@ubuntu:~S
```

ALLOCATING MEMORY TO A PROGRAM

CODE: <u>allocate.c</u>

SCREENSHOTS

```
vmdhruv@ubuntu:~$ gedit allocate.c &
[1] 2858
vmdhruv@ubuntu:~$ gcc -o allocate allocate.c
                               gedit allocate.c
[1]+ Done
vmdhruv@ubuntu:~$ ./allocate
Allocated 1 MB
Allocated 2 MB
Allocated 3 MB
Allocated 4 MB
Allocated 5 MB
Allocated 6 MB
Allocated 7 MB
Allocated 8 MB
Allocated 9 MB
Allocated 10 MB
Allocated 11 MB
Allocated 12 MB
Allocated 13 MB
Allocated 14 MB
Allocated 15 MB
Allocated 16 MB
Allocated 17 MB
Allocated 18 MB
Allocated 19 MB
Allocated 20 MB
```

The program gets killed eventually, before it reaches its desired memory allocation.

```
Allocated 1140 MB
Allocated 1141 MB
Allocated 1142 MB
Allocated 1143 MB
Allocated 1144 MB
Allocated 1145 MB
Allocated 1146 MB
Allocated 1147 MB
Allocated 1148 MB
Allocated 1149 MB
Allocated 1150 MB
Allocated 1151 MB
Allocated 1152 MB
Allocated 1153 MB
Allocated 1154 MB
Allocated 1155 MB
Allocated 1156 MB
Allocated 1157 MB
Allocated 1158 MB
Allocated 1159 MB
Allocated 1160 MB
Allocated 1161 MB
Killed
vmdhruv@ubuntu:~$
```

TESTING THE TIME TAKEN FOR SEQUENTIAL FILE ACCESS AND RANDOM FILE ACCESS

CODE: randomSeqAccess.cpp

```
#include <iostream>
                                                                    char temp;
#include <fstream>
                                                                    bool used = false, found =false;
#include <time.h>
                                                                    int pos = 0, n = 0;
#include <stdlib.h>
                                                                    while(n < size-1)
#include <exception>
                                                                    {
using namespace std;
                                                                      found = false;
                                                                      do
void readseq() // sequential read
{
                                                                         used = false;
  ifstream fin;
                                                                         pos = rand()\%(size-1);
  fin.open("notes.txt");
                                                                         for (int i = 0; i < n; ++i)
  char temp;
  while(fin.get(temp))
                                                                           if(a[i] == pos)
    //cout << temp;
                                                                           {
  //cout << endl;
                                                                             used = true;
  fin.close();
                                                                             break;
}
                                                                           }
void randomseek() // random seek
                                                                         }
                                                                        if(!used)
  srand((double)(clock())); // seed rand with
current time
                                                                           found = true;
  ifstream fin;
                                                                           a[n] = pos;
  fin.open("notes.txt");
                                                                           n++;
  fin.seekg(0, ios::end);
                                                                         }
  int size = fin.tellg();
                                                                      }
  fin.clear();
                                                                      while(!found);
  int *a;
                                                                        fin.seekg(a[n-1], ios::beg);
  //cout<<size<<endl;
                                                                      fin.get(temp);
  try
                                                                      //cout <<temp;</pre>
                                                                    }
    a = new int[size];
                                                                    //cout << endl;
                                                                    delete a;
  catch(bad alloc& ba)
                                                                    fin.close();
    cerr << "Bad alloc caught: " << ba.what() << endl;</pre>
  }
```

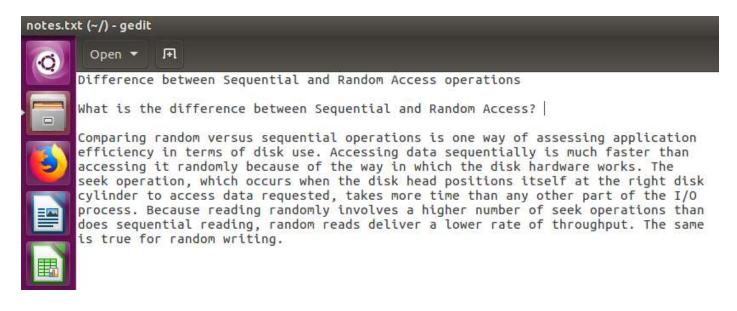
```
void timeit( void (*func)(void) )
                                                               int main()
{
                                                               {
  double time_taken;
                                                                  cout << "\n\nReading file => notes.txt (SEQUENTIAL
                                                               ACCESS):\n";
  cout << "Starting process...\n";</pre>
                                                                  timeit(readseq);
  clock_t timer = clock();
                                                                  cout << "\n\nReading file => notes.txt (RANDOM
  (*func)();
                                                               ACCESS):\n";
  time taken = (clock() -
                                                                  timeit(randomseek);
timer)/(double)CLOCKS_PER_SEC;
                                                                  return 0;
  cout << "Time taken: " << time taken << endl;
}
```

SCENARIO 1: Reading a file named "dumpfile2"

```
dumpfile2 (~/) - gedit
                  ıП
        Open 🔻
         -815968130 -1157655052 -1820109930 -1909213681
        -1537236168
                     1613725304
                                  1616991439
                                               1428695555
         1060375142 -1876861793
                                   171258402
                                               1609491991
        -1615945021
                      535262009
                                  1827393893
                                               -305230085
         -521017845
                      786042693
                                  2145066524
                                               1439556792
                      633221205
                                   168737306 -1547173034
         1528464545
         1537190387
                      471835489
                                 1598679725
                                               -271775949
         1677028527
                     2137285519
                                  -248449310
                                               1904786526
         -148389002
                     2105190240
                                   835216135
                                                 30194786
         1898616510
                      566206500
                                   299800318
                                               -913749864
          275759874
                      446608638
                                  1430123793
                                               1886675438
         -232127153 -2140945209
                                  -465655335
                                               2122644243
                     -724263485 -1226142826 -1051771122
         1437122677
          304187994
                       -36906097 -1729751422
                                                 34538065
                                 1551559410
         1761605234 -1718702726
                                                946194828
         -757186220 -2104005227
                                   290335313 -1700406980
         1943690607
                       59539195
                                   248963579
                                               -149411599
        -1477407591
                       87619201 -1223750149 -1528502171
         1794089912
                     -394959653 -1934241948
                                               -692377336
          788428720
                     -605835377
                                  -794461777
                                                 99526346
        -1402252698
                     2110711648
                                   210191276
                                                524853416
         2027217871
                     1504744744 - 1947938880
                                                721113442
        -1409770638
                     -376697956
                                   822349493
                                               -417269620
          856715352
                       63334070 -2002135136 -1813866857
          394049559
                     2010767204 -559222527
                                                843865281
         -642396977
                     -714791469
                                 -535887575 -1106848208
         -485820193
                                               1912555449
                     1375523182
                                  -341809184
        -1824072893
                     -724392419
                                   787887978
                                                 18744614
         -690079488 -1981225115 -1743509943
                                                748651628
```

Checking the sequential and random access time

SCENARIO 2: Reading a file named "notes.txt"



Checking the sequential and random access time