David Harris COMP305 - Assingment 2

Question 10.2

The operating system should have one user-wide open file table. When two processes open the same file, the system-wide file table has two entries, one for each instance of the open file.

Question 11.1

a) If all extents are the same size then the advantages of using extents are maximised. The use of extents means that the internal and external fragmentation are kept to a minimum. This is because rather than the file needed to be spread across contiguous blocks, it can be spread across multiple extents that are not necessarily contiguous.
b) If the extent size is too large, it will lead to increased internal fragmentation because a 'hole' may not be big enough for the extent to fit in, so it has to search for a place to fit the large extent.
c) If the extent size is too small, external fragmentation will increase, which will decrease overall performance.

Question 11.6

a) Contiguous
   1. Contiguous blocks are blocks that are laid out next to each other, one after the other. We can calculate the location of the nth block given the first one.
   2. Only one block has to be read, because we can calculate the location of the 4th block given the location of the 10th block.
b) Linked
   1. Linked blocks means that each block forms a linked list, with data and a link to the next block. The final blocks' pointer is null.
   2. For direct access, it takes n reads, so in this case, we need to read 6 blocks to get to block 4.
c) Indexed
   1. Indexed blocks means that a single block has a table that holds the rest of the blocks that make up the file. Each block is therefore data-only.
   2. As the block size is less than 512, I assume that the file is in sequential mode, rather than direct mode. Therefore, we must do 6 reads in order to read block 6.

Question 12.1

a) All disk-scheduling algorithms apart from FCFS are non-fair because they allow requests to be made ahead of the next request in the queue. This means that if requests are made in such a way, the next request may never be serviced. FCFS doesn't have this problem because each request is guaranteed to be serviced. LOOK and SCAN also tend to favour the middle sectors, and thus will promote unfairness.
b) An age timer could be used to make sure that each request is seen to in a timely manner. This would decrease performance greatly however, if the head was forced to read a block that was outside the algorithm's calculated order.

c) Fairness for a time-sharing system is important because with multiple users making requests, you want to make sure that jobs requests get done, and are not sitting waiting for newer requests for a long time.

d) A database server is somewhere where unfairness could be good. A database server needs to have very quick reads and writes. If an unfair I/O requests were made on locked tables, then performance would not drop.
A web server could also handle unfair I/O requests. The webserver will give out the main page all the time, so the performance will be fast for that. For 'off the beaten track' pages, performance will be slower, but it won't matter too much.

Question 12.2

SCSF

Seek	Difference
143	
86	57
86	
1470	1384
1470	
913	557
913	
1774	861
1774	
948	826
948	
1509	561
1509	
1022	487
1022	
1750	728
1750	
130	1620
Total	6322

SSFT

Seek	Difference
143	
130	13
130	
86	44
86	
913	827
913	
948	35
948	
1022	74
1022	
1470	448
1470	
1509	39
1509	
1750	241
1750	
1774	24
Total	1745

SCAN

Seek	Difference
143	
130	13
130	
86	44
86	
0	86
0	
913	913
913	
948	35
948	
1022	74
1022	
1470	448
1470	
1509	39
1509	
1774	265
Total	1917

LOOK

Seek	Difference
143	
130	13
130	
86	44
86	
913	827
913	
948	35
948	
1022	74
1022	
1470	448
1470	
1509	39
1509	
1750	241
1750	
1774	24
Total	1745

C-SCAN

Seek	Difference
143	
913	770
913	
948	35
948	
1022	74
1022	
1470	448
1470	
1509	39
1509	
1750	241
1750	
1774	24
1774	
1509	265
1509	
4999	3490
4999	
0	4999
0	
86	86
86	
130	44
Total	5386