Assignment 2

Adnan Altukleh
Abdulkarim Dawalibi

Home assignment 3
The size of allocated memory is 4294967296 byte
around 1 GB for each iteration is written to the temporary file

Home assignment 4
In swpd,buff and cache columns you can find the amount of memory a process uses
In si and so you can find the number of swap ins and outs
In bi and bo you can find the number of I/O blocks read and written

cpu time can find in “TIME+” column, the cpu utilization can found in %CPU with help of top command

Task 1
memory does the program use 4197 MB we check it with help of vmstat “free”, yes
cpu utilization when executing the test1 program CPU utilization 99,7, we check it with help of top
The type of program is CPU bound

Around 1 GB memory does the program use, we check it with help of vmstat “free”
Bo: 4891964 out blocks
cpu utilization when executing the test1 program CPU utilization around 61,14, we check it with help of top.
The type of program is I/O bound

Task 2
The difference between them run1 execute test1 3 times then execute test2 2 times sequentially
where the run2 execute test1 3 times and test2 2 times but parallelly

---

the execution time for run1 1m56,927s
when we executed run1 started with test1 program the cpu utilization was around 24-20 but when the test2 started to executed the utilization was around 57-55

When we executed run2 the cpu utilization was around 77-60, when test2 was done executing the cpu utilization went down to 25-20.
the execution time for run2: 1m 21,939s

The run2 executes faster and utilization is best where the value is higer which is in run2 case.

Task 3
We implemented a fifo queue and iterated through all references in mem file with the following parameter, Number of physical pages, page size and the file name.

Task 4

mp3d.mem

Number of pages

page size	1	2	4	8	16	32	64	128
128	55421	22741	13606	6810	3121	1503	1097	877
256	54357	20395	11940	4845	1645	939	669	478
512	52577	16188	9458	2372	999	629	417	232
1024	51804	15393	8362	1330	687	409	193	99

mult.mem

Number of pages

page size	1	2	4	8	16	32	64	128
128	45790	22303	18034	1603	970	249	67	67
256	45725	22260	18012	1529	900	223	61	61
512	38246	16858	2900	1130	489	210	59	59
1024	38245	16855	2890	1124	479	204	57	57

Task 5

1. The number of page faults decrease, because we expend the interval of address reference
2. The number of page faults decrease, because we make more memory available
3. Doubling the frame size “number of pages” decreases the page faults,wheel halving the number of pages “frame size” causes an increment of the page faults.The reason we have less number of pages at the same time we have a large interval.

4. The point where the page faults decrease drastically are when (page size,number of pages) (512,4) and (1024,4), the reason is that the page size is larger which increase the interval
5. The point where the number of pages is 64 and 128 doesn't decrease anymore, because of the continuous interval so we don't have an empty space to have a new page fault.

Task 6

We implemented a LRU and iterated through all references in mem file with the following parameter, Number of physical pages, page size and the file name.

Task 7

mp3d.mem

	Number of pages

page size	1	2	4	8	16	32	64	128
128	55421	16973	11000	6536	1907	995	905	796
256	54357	14947	9218	3811	794	684	577	417
512	52577	11432	6828	1617	603	503	362	206
1024	51804	10448	5605	758	472	351	167	99

Task 8

1. The LRU performs better than FIFO where the number of pages is higher than 1, because LRU keeps the most appeared references in the queue while FIFO doesn't consider that.
2. when the number of pages is 1, because we have just one page and in this case both algorithms are going to work similarly

Task 9

We implemented an Optimal page replacement (Bélády's algorithm) and iterated through all references in mem file with the following parameter, Number of physical pages, page size and the file name.

Task 10
mp3d.mem

	Number of pages

page size	1	2	4	8	16	32	64	128
128	55421	15856	8417	3565	1092	824	692	558
256	54357	14168	6431	1919	652	517	395	295
512	52577	11322	4191	920	470	340	228	173
1024	51804	10389	3367	496	339	213	107	99

Task 11

1. Because the algorithm deletes the frame that has the minimum chance to appear again in the near future.
2. Because it is very hard to predict the future.
3. Yes in page size 128,256,512,1024 and number of pages 1, Because the Frames size is just 1 therefore, the algorithms will work similarly.
   There is another point also when the page size is 1024 and the number of pages is 128,Because we get a space for the same number of pages with this page size and number of pages for all algorithms.