Lab-10 Report

Group Members:

130050007 : Vaibhav Bhosale
130050046 : Dibyendu Mondal

Part A:

Command used: pmap -x 12086

Question 1:

```
12086:   ./A
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       4       4       0 r-x-- A
0000000000600000       4       4       4 r---- A
0000000000601000       4       4       4 rw--- A
00007f165fa10000    1792    1020       0 r-x-- libc-2.21.so
00007f165fbd0000    2048       0       0 ----- libc-2.21.so
00007f165fdd0000      16      16      16 r---- libc-2.21.so
00007f165fdd4000       8       8       8 rw--- libc-2.21.so
00007f165fdd6000      16      12      12 rw---   [ anon ]
00007f165fdda000     144     144       0 r-x-- ld-2.21.so
00007f165ffd2000      12      12      12 rw---   [ anon ]
00007f165fff9000      16      12      12 rw---   [ anon ]
00007f165fffd000       4       4       4 r---- ld-2.21.so
00007f165fffe000       4       4       4 rw--- ld-2.21.so
00007f165ffff000       4       4       4 rw---   [ anon ]
00007ffca255e000     132      12      12 rw---   [ stack ]
00007ffca25b6000       8       0       0 r----   [ anon ]
00007ffca25b8000       8       4       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 r-x--   [ anon ]
---------------- ------- ------- -------
total kB           4228    1264      92
```

VmSize: 4228 kB
VmRSS: 1264 kB

Question 2:

```
12086:   ./A
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       4       4       0 r-x-- A
0000000000600000       4       4       4 r---- A
```

```
0000000000601000       4       4       4 rw--- A
00007f165f010000   10240       0       0 rw-s- foo0.txt
00007f165fa10000    1792    1020       0 r-x-- libc-2.21.so
00007f165fbd0000    2048       0       0 ----- libc-2.21.so
00007f165fdd0000      16      16      16 r---- libc-2.21.so
00007f165fdd4000       8       8       8 rw--- libc-2.21.so
00007f165fdd6000      16      12      12 rw---   [ anon ]
00007f165fdda000     144     144       0 r-x-- ld-2.21.so
00007f165ffd2000      12      12      12 rw---   [ anon ]
00007f165fff9000      16      16      16 rw---   [ anon ]
00007f165fffd000       4       4       4 r---- ld-2.21.so
00007f165fffe000       4       4       4 rw--- ld-2.21.so
00007f165ffff000       4       4       4 rw---   [ anon ]
00007ffca255e000     132      12      12 rw---   [ stack ]
00007ffca25b6000       8       0       0 r----   [ anon ]
00007ffca25b8000       8       4       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 r-x--   [ anon ]
---------------- ------- ------- -------
total kB           24708    1268      96

VmSize:            14468 kB
VmRSS:              1268 kB
```

The virtual memory allocated to the process increases by 10MB, thus incorporating the mapping of the file(10MB) onto the virtual memory. VmRSS increases as some library is loaded.
The mapping is created in the virtual memory space, therefore the memory resident in the physical RAM won't change.

Question 3:

```
12086:   ./A
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       4       4       0 r-x-- A
0000000000600000       4       4       4 r---- A
0000000000601000       4       4       4 rw--- A
00007f165f010000   10240      64       0 rw-s- foo0.txt
00007f165fa10000    1792    1084       0 r-x-- libc-2.21.so
00007f165fbd0000    2048       0       0 ----- libc-2.21.so
00007f165fdd0000      16      16      16 r---- libc-2.21.so
00007f165fdd4000       8       8       8 rw--- libc-2.21.so
00007f165fdd6000      16      12      12 rw---   [ anon ]
00007f165fdda000     144     144       0 r-x-- ld-2.21.so
00007f165ffd2000      12      12      12 rw---   [ anon ]
00007f165fff9000      16      16      16 rw---   [ anon ]
00007f165fffd000       4       4       4 r---- ld-2.21.so
```

```
00007f165fffe000       4       4       4 rw--- ld-2.21.so
00007f165ffff000       4       4       4 rw---   [ anon ]
00007ffca255e000     132      12      12 rw---   [ stack ]
00007ffca25b6000       8       0       0 r----   [ anon ]
00007ffca25b8000       8       4       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 r-x--   [ anon ]
---------------- ------- ------- -------
total kB           24708    1396      96

VmSize:            14468 kB
VmRSS:              1396 kB
```

The virtual memory allocated remains the same, since now the file is completely loaded on the virtual memory and available as an array to the process. VmRSS increases by 64 kB as a file is loaded on accessing the 1st character.

Question 4:

```
12086:   ./A
Address           Kbytes     RSS   Dirty Mode  Mapping
0000000000400000       4       4       0 r-x-- A
0000000000600000       4       4       4 r---- A
0000000000601000       4       4       4 rw--- A
00007f165f010000   10240      64       0 rw-s- foo0.txt
00007f165fa10000    1792    1084       0 r-x-- libc-2.21.so
00007f165fbd0000    2048       0       0 ----- libc-2.21.so
00007f165fdd0000      16      16      16 r---- libc-2.21.so
00007f165fdd4000       8       8       8 rw--- libc-2.21.so
00007f165fdd6000      16      12      12 rw---   [ anon ]
00007f165fdda000     144     144       0 r-x-- ld-2.21.so
00007f165ffd2000      12      12      12 rw---   [ anon ]
00007f165fff9000      16      16      16 rw---   [ anon ]
00007f165fffd000       4       4       4 r---- ld-2.21.so
00007f165fffe000       4       4       4 rw--- ld-2.21.so
00007f165ffff000       4       4       4 rw---   [ anon ]
00007ffca255e000     132      12      12 rw---   [ stack ]
00007ffca25b6000       8       0       0 r----   [ anon ]
00007ffca25b8000       8       4       0 r-x--   [ anon ]
ffffffffff600000       4       0       0 r-x--   [ anon ]
---------------- ------- ------- -------
total kB           24708    1396      96

VmSize:            14468 kB
VmRSS:              1396 kB
```

The virtual memory allocated remains the same, since now the file is completely loaded on the virtual memory and available as an array to the process. Therefore, to access the character at an offset 10000 bytes(basically the 10000th character in the file), nothing new needs to loaded onto the virtual memory .
The character would be accessed using the already loaded page, hence no change in the VM-RSS part.

Part B:

Question 1:

Before:

	total	used	free	shared	buffers	cached
Mem:	8065444	4650880	3414564	481748	1080	863816
-/+ buffers/cache:		3785984	4279460			
Swap:	7813116	0	7813116			

After:

	total	used	free	shared	buffers	cached
Mem:	8065444	4911820	3153624	481952	11980	1143336
-/+ buffers/cache:		3756504	4308940			
Swap:	7813116	0	7813116			

Throughput 43.507195 MB/s
The bottleneck in this case is the disk (checked using iostat).

Question 2:

Before:

	total	used	free	shared	buffers	cached
Mem:	8065444	4625444	3440000	482368	1224	864460
-/+ buffers/cache:		3759760	4305684			
Swap:	7813116	0	7813116			

After:

	total	used	free	shared	buffers	cached
Mem:	8065444	4955776	3109668	482272	12216	1165920
-/+ buffers/cache:		3777640	4287804			
Swap:	7813116	0	7813116			

Throughput 37.689921 MB/s
The bottleneck in this case is the disk (checked using iostat).

Question 3:

No, using memory-mapped files do not give any added performance benefits over regular files. We get roughly the same value of throughput in both the cases.

Question 4:

The disk buffer cache is used in the same way when reading from regular and memory-mapped files. Therefore, the difference before and after in both the cases is roughly the same.

Question 5:

Before:

	total	used	free	shared	buffers	cached
Mem:	8065444	4915440	3150004	515216	1248	902112
-/+ buffers/cache:		4012080	4053364			
Swap:	7813116	0	7813116			

After:

	total	used	free	shared	buffers	cached
Mem:	8065444	5206768	2858676	516720	5052	1164108
-/+ buffers/cache:		4037608	4027836			
Swap:	7813116	0	7813116			

Throughput 16.036659 MB/s
The bottleneck is disc (checked using iostat).

Question 6:

Before:

	total	used	free	shared	buffers	cached
Mem:	8065444	5040248	3025196	521604	2044	908648
-/+ buffers/cache:		4129556	3935888			
Swap:	7813116	0	7813116			

After:

	total	used	free	shared	buffers	cached
Mem:	8065444	5310232	2755212	521620	5784	1167444
-/+ buffers/cache:		4137004	3928440			
Swap:	7813116	0	7813116			

Throughput 34.076821 MB/s
The bottleneck is disc (checked using iostat).

Question 7:

The throughput in case of memory-mapped files is lesser than that of regular files. The reason being that when writing to memory mapped files, we need to run the “msync” function to ensure that the changes are written on the actual location of the file, as we used MAP_SHARED.

Question 8:

In case of Memory map:
Throughput 37.852549 MB/s

In case of Regular Files:
Throughput 294.565271 MB/s

No, memory-mapped files do not give any performance benefits over regular files for frequent writes. On the contrary, it performs worse than regular files giving poor throughput values. Memory-mapping does not utilise the disk buffer cache to the extent utilised by the regular files.