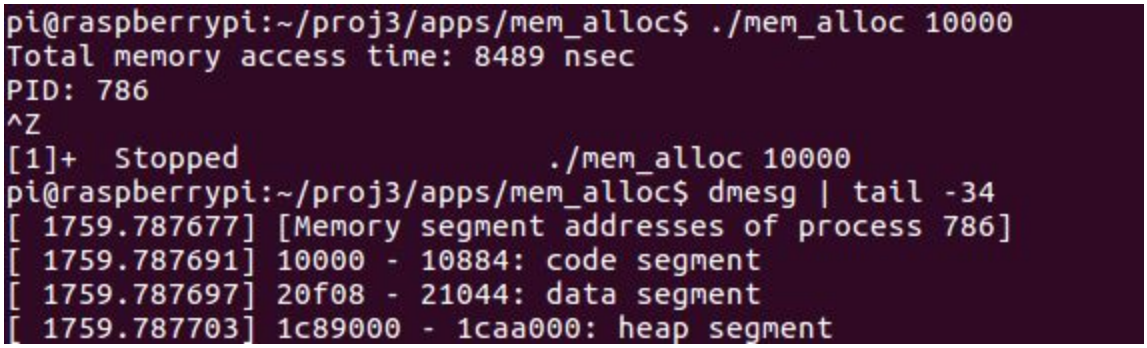
1. What does 'mlockall(MCL_CURRENT | MCL_FUTURE)' do?
    a. The flag MCL_CURRENT causes all currently mapped pages to stay locked on the physical memory while the MCL_FUTURE flag causes all pages that will be mapped or new pages to be locked in physical memory. This essentially causes all memory to be locked.
2. Report the memory access time of mem_alloc and mem_alloc_lock (assignment 4.1 and 4.2) on Raspberry Pi 3 with the memory size of 10 KB, 1 MB, and 100MB. Compute average memory access time for each case.

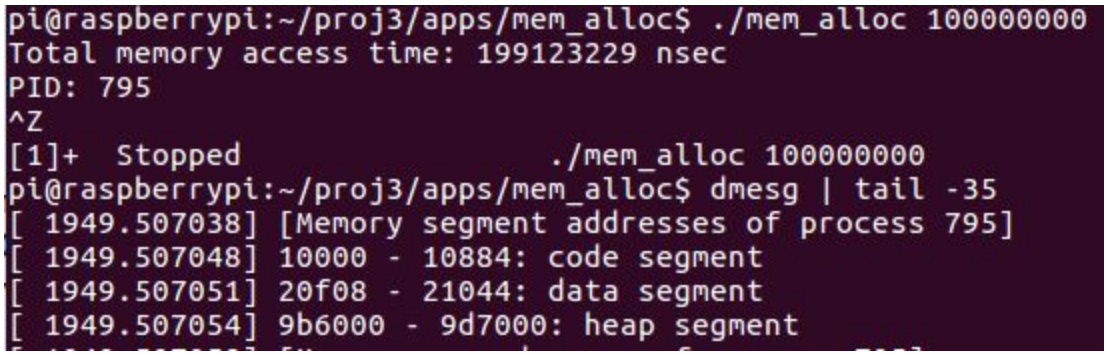
```
pi@raspberrypi:~/proj3/apps/mem_alloc $ ./mem_alloc 10000
Total memory access time: 9062 nsec
PID: 1257
^C
pi@raspberrypi:~/proj3/apps/mem_alloc $ ./mem_alloc 1000000
Total memory access time: 2029164 nsec
PID: 1258
^C
pi@raspberrypi:~/proj3/apps/mem_alloc $ ./mem_alloc 100000000
Total memory access time: 183978019 nsec
```

```
pi@raspberrypi:~/proj3/apps/mem_alloc_lock $ ./mem_alloc_lock 10000
Total memory access time: 1406 nsec
PID: 1285
^[[A^[[A^C
pi@raspberrypi:~/proj3/apps/mem_alloc_lock $ ./mem_alloc_lock 1000000
Total memory access time: 20625 nsec
PID: 1286
^C
pi@raspberrypi:~/proj3/apps/mem_alloc_lock $ sudo ./mem_alloc_lock 100000000
Total memory access time: 1873596 nsec
PID: 1292
```

	mem_alloc					
Size(Bytes)	Run 1 (ns)	Run 2 (ns)	Run 3 (ns)	Run 4 (ns)	Average Total Time(ns)	Average Time per Byte(ns)
10000	9062	8645	8679	11458	9461	0.9461
1000000	2029164	2283727	2014991	2022284	2087541.5	2.0875415
100000000	183978019	178951822	177424078	177888026	179560486.3	1.795604863
	mem_alloc_lock					
Size(Bytes)	Run 1 (ns)	Run 2 (ns)	Run 3 (ns)	Run 4 (ns)	Average Total Time(ns)	Average Time per Byte(ns)
10000	1406	1354	1302	1354	1354	0.1354
1000000	20625	20104	19999	20364	20273	0.020273
100000000	1873596	1973761	1798350	1807777	1863371	0.01863371

3. Report the kernel logs of the show_segment_info() syscall (assignment 4.3) for mem_alloc (assignment 4.1) with the memory size of 10 KB and 100MB. Also, report the size of each segment. (Note: you may see somewhat counter-intuitive results. Read Section 6. Q&A).





	Code		Data		Heap	
mem_alloc	**Hex**	**Dec**	**Hex**	**Dec**	**Hex**	**Dec**
10000	884	2180	13C	316	21000	135168
100000000	884	2180	13C	316	1000	4096

4. If the OS kernel uses virtual memory with demand paging but does not provide mlock-like functions, then what can be a workaround that a user-level program can do in order to prevent unpredictable delay in memory access at runtime?
    a. Demand paging works by only copying a page into physical memory when it is necessary or "in demand." A way to keep a page "in demand" is by constantly recreating the process. This can be done by periodically creating child processes to synthetically create "demand."