REAL-TIME EMBEDDED SYSTEMS (EE_255_001_24W, CS_251_001_24W)

HW3 - TEAM 06

Keerthana Sunil Babu Bidare - kbida003
Dhanush Radhakrishna - dradh003
Akshay Jayaram - ajaya026

Question 1:

Memory access time:

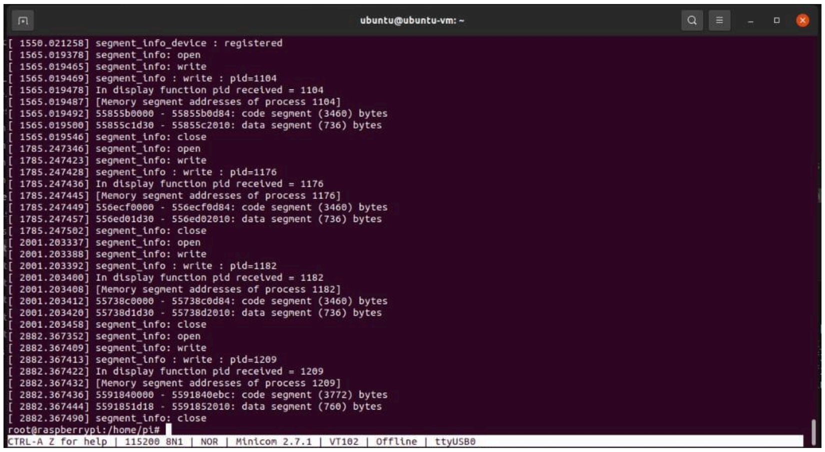
	mem_alloc	**mem_alloc_lock**
1MB	1847315 ns	9203 ns
10MB	17649056 ns	227259 ns
100MB	138772222 ns	1159259 ns

The table shows a significant performance difference between mem_alloc and mem_alloc_lock for memory allocation.

- Faster Execution with Locking: For all memory sizes (1MB, 10MB, 100MB), mem_alloc_lock is considerably faster than mem_alloc, indicating that the overhead of locking in mem_alloc_lock is negligible compared to the overall process of memory allocation in mem_alloc. After allocating the bytes in mem_alloc_lock the memory is locked. This avoids swapping of pages between physical and virtual memory, hence there is no latency by page fault while writing to the memory location resulting in less memory access time in mem_alloc_lock compared to mem_alloc.
- Increasing Speedup: The speedup of mem_alloc_lock over mem_alloc increases as the memory size grows. This indicates that the additional overhead of mem_alloc becomes more prominent for larger allocations.

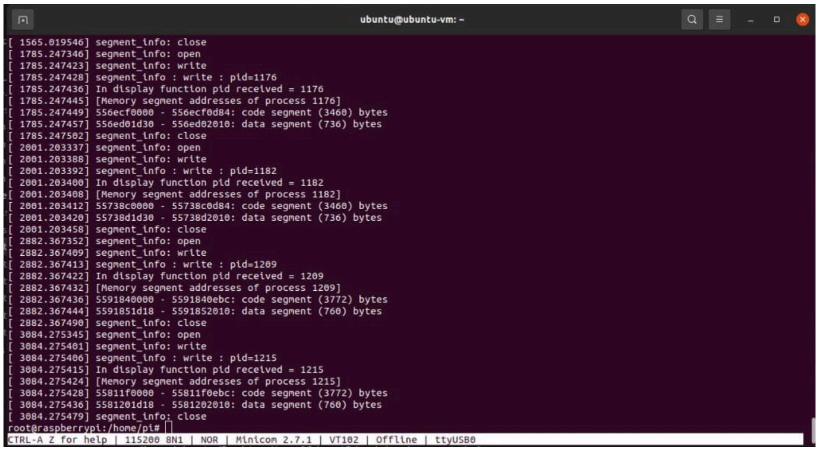
Question 2:

Kernel logs of segment_info:
1KB



```
[ 1550.021258] segment_info_device : registered
[ 1565.019378] segment_info: open
[ 1565.019465] segment_info: write
[ 1565.019469] segment_info : write : pid=1104
[ 1565.019478] In display function pid received = 1104
[ 1565.019487] [Memory segment addresses of process 1104]
[ 1565.019492] 55855b0000 - 55855b0d84: code segment (3460) bytes
[ 1565.019500] 55855c1d30 - 55855c2010: data segment (736) bytes
[ 1565.019546] segment_info: close
[ 1785.247346] segment_info: open
[ 1785.247423] segment_info: write
[ 1785.247428] segment_info : write : pid=1176
[ 1785.247436] In display function pid received = 1176
[ 1785.247445] [Memory segment addresses of process 1176]
[ 1785.247449] 556ecf0000 - 556ecf0d84: code segment (3460) bytes
[ 1785.247457] 556ed01d30 - 556ed02010: data segment (736) bytes
[ 1785.247502] segment_info: close
[ 2001.203337] segment_info: open
[ 2001.203388] segment_info: write
[ 2001.203392] segment_info : write : pid=1182
[ 2001.203400] In display function pid received = 1182
[ 2001.203408] [Memory segment addresses of process 1182]
[ 2001.203412] 55738c0000 - 55738c0d84: code segment (3460) bytes
[ 2001.203420] 55738d1d30 - 55738d2010: data segment (736) bytes
[ 2001.203458] segment_info: close
[ 2882.367352] segment_info: open
[ 2882.367409] segment_info: write
[ 2882.367413] segment_info : write : pid=1209
[ 2882.367422] In display function pid received = 1209
[ 2882.367432] [Memory segment addresses of process 1209]
[ 2882.367436] 5591840000 - 5591840ebc: code segment (3772) bytes
[ 2882.367444] 5591851d18 - 5591852010: data segment (760) bytes
[ 2882.367490] segment_info: close
root@raspberrypi:/home/pi# 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

100MB

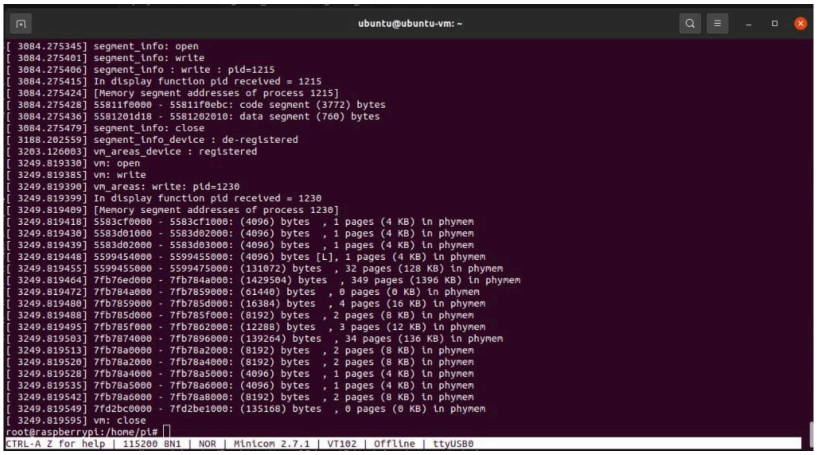


```
[ 1565.019546] segment_info: close
[ 1785.247346] segment_info: open
[ 1785.247423] segment_info: write
[ 1785.247428] segment_info : write : pid=1176
[ 1785.247436] In display function pid received = 1176
[ 1785.247445] [Memory segment addresses of process 1176]
[ 1785.247449] 556ecf0000 - 556ecf0d84: code segment (3460) bytes
[ 1785.247457] 556ed01d30 - 556ed02010: data segment (736) bytes
[ 1785.247502] segment_info: close
[ 2001.203337] segment_info: open
[ 2001.203388] segment_info: write
[ 2001.203392] segment_info : write : pid=1182
[ 2001.203400] In display function pid received = 1182
[ 2001.203408] [Memory segment addresses of process 1182]
[ 2001.203412] 55738c0000 - 55738c0d84: code segment (3460) bytes
[ 2001.203420] 55738d1d30 - 55738d2010: data segment (736) bytes
[ 2001.203458] segment_info: close
[ 2882.367352] segment_info: open
[ 2882.367409] segment_info: write
[ 2882.367413] segment_info : write : pid=1209
[ 2882.367422] In display function pid received = 1209
[ 2882.367432] [Memory segment addresses of process 1209]
[ 2882.367436] 5591840000 - 5591840ebc: code segment (3772) bytes
[ 2882.367444] 5591851d18 - 5591852010: data segment (760) bytes
[ 2882.367490] segment_info: close
[ 3084.275345] segment_info: open
[ 3084.275401] segment_info: write
[ 3084.275406] segment_info : write : pid=1215
[ 3084.275415] In display function pid received = 1215
[ 3084.275424] [Memory segment addresses of process 1215]
[ 3084.275428] 55811f0000 - 55811f0ebc: code segment (3772) bytes
[ 3084.275436] 5581201d18 - 5581202010: data segment (760) bytes
[ 3084.275479] segment_info: close
root@raspberrypi:/home/pi# 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

Question 3:

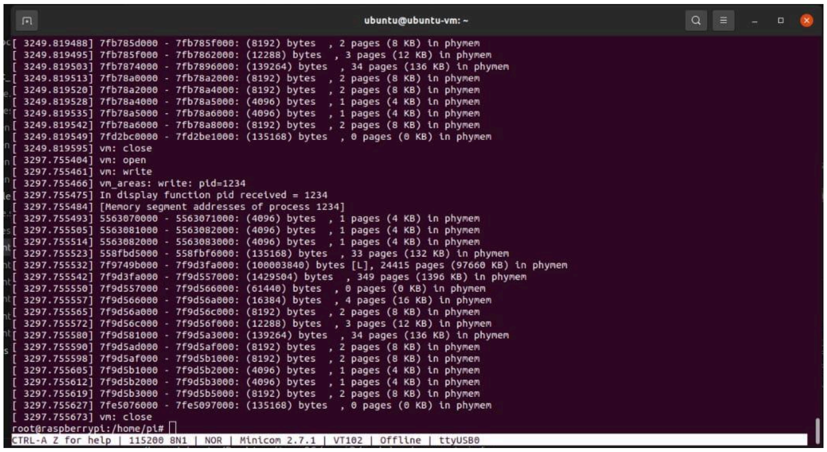
Kernel logs of vm_areas:

1KB



```
[ 3084.275345] segment_info: open
[ 3084.275401] segment_info: write
[ 3084.275406] segment_info : write : pid=1215
[ 3084.275415] In display function pid received = 1215
[ 3084.275424] [Memory segment addresses of process 1215]
[ 3084.275428] 55811f0000 - 55811f0ebc: code segment (3772) bytes
[ 3084.275436] 5581201d18 - 5581202010: data segment (760) bytes
[ 3084.275479] segment_info: close
[ 3188.202559] segment_info_device : de-registered
[ 3203.126003] vm_areas_device : registered
[ 3249.819330] vm: open
[ 3249.819385] vm: write
[ 3249.819390] vm_areas: write: pid=1230
[ 3249.819399] In display function pid received = 1230
[ 3249.819409] [Memory segment addresses of process 1230]
[ 3249.819418] 5583cf0000 - 5583cf1000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819430] 5583d01000 - 5583d02000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819439] 5583d02000 - 5583d03000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819448] 5599454000 - 5599455000: (4096) bytes [L], 1 pages (4 KB) in phymem
[ 3249.819455] 5599455000 - 5599475000: (131072) bytes  , 32 pages (128 KB) in phymem
[ 3249.819464] 7fb76ed000 - 7fb784a000: (1429504) bytes  , 349 pages (1396 KB) in phymem
[ 3249.819472] 7fb784a000 - 7fb7859000: (61440) bytes  , 0 pages (0 KB) in phymem
[ 3249.819480] 7fb7859000 - 7fb785d000: (16384) bytes  , 4 pages (16 KB) in phymem
[ 3249.819488] 7fb785d000 - 7fb785f000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819495] 7fb785f000 - 7fb7862000: (12288) bytes  , 3 pages (12 KB) in phymem
[ 3249.819503] 7fb7874000 - 7fb7896000: (139264) bytes  , 34 pages (136 KB) in phymem
[ 3249.819513] 7fb78a0000 - 7fb78a2000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819520] 7fb78a2000 - 7fb78a4000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819528] 7fb78a4000 - 7fb78a5000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819535] 7fb78a5000 - 7fb78a6000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819542] 7fb78a6000 - 7fb78a8000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819549] 7fd2bc0000 - 7fd2be1000: (135168) bytes  , 0 pages (0 KB) in phymem
[ 3249.819595] vm: close
root@raspberrypi:/home/pi# 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

100MB



```
[ 3249.819488] 7fb785d000 - 7fb785f000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819495] 7fb785f000 - 7fb7862000: (12288) bytes  , 3 pages (12 KB) in phymem
[ 3249.819503] 7fb7874000 - 7fb7896000: (139264) bytes  , 34 pages (136 KB) in phymem
[ 3249.819513] 7fb78a0000 - 7fb78a2000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819520] 7fb78a2000 - 7fb78a4000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819528] 7fb78a4000 - 7fb78a5000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819535] 7fb78a5000 - 7fb78a6000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3249.819542] 7fb78a6000 - 7fb78a8000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3249.819549] 7fd2bc0000 - 7fd2be1000: (135168) bytes  , 0 pages (0 KB) in phymem
[ 3249.819595] vm: close
[ 3297.755404] vm: open
[ 3297.755461] vm: write
[ 3297.755466] vm_areas: write: pid=1234
[ 3297.755475] In display function pid received = 1234
[ 3297.755484] [Memory segment addresses of process 1234]
[ 3297.755493] 5563070000 - 5563071000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3297.755505] 5563081000 - 5563082000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3297.755514] 5563082000 - 5563083000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3297.755523] 558fbd5000 - 558fbf6000: (135168) bytes  , 33 pages (132 KB) in phymem
[ 3297.755532] 7f9749b000 - 7f9d3fa000: (100003840) bytes [L], 24415 pages (97660 KB) in phymem
[ 3297.755542] 7f9d3fa000 - 7f9d557000: (1429504) bytes  , 349 pages (1396 KB) in phymem
[ 3297.755550] 7f9d557000 - 7f9d566000: (61440) bytes  , 0 pages (0 KB) in phymem
[ 3297.755557] 7f9d566000 - 7f9d56a000: (16384) bytes  , 4 pages (16 KB) in phymem
[ 3297.755565] 7f9d56a000 - 7f9d56c000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3297.755572] 7f9d56c000 - 7f9d56f000: (12288) bytes  , 3 pages (12 KB) in phymem
[ 3297.755580] 7f9d581000 - 7f9d5a3000: (139264) bytes  , 34 pages (136 KB) in phymem
[ 3297.755590] 7f9d5ad000 - 7f9d5af000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3297.755598] 7f9d5af000 - 7f9d5b1000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3297.755605] 7f9d5b1000 - 7f9d5b2000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3297.755612] 7f9d5b2000 - 7f9d5b3000: (4096) bytes  , 1 pages (4 KB) in phymem
[ 3297.755619] 7f9d5b3000 - 7f9d5b5000: (8192) bytes  , 2 pages (8 KB) in phymem
[ 3297.755627] 7fe5076000 - 7fe5097000: (135168) bytes  , 0 pages (0 KB) in phymem
[ 3297.755673] vm: close
root@raspberrypi:/home/pi# 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

Question 4:

In the absence of mlock functions, a user-level program can tackle unpredictable memory access delays caused by demand paging in a few ways. It can pre-fetch anticipated data, track and prioritize recently used pages, allocate larger memory space, or proactively touch future memory regions. Additionally, reducing the program's overall memory footprint helps. While these workarounds add complexity and have limitations, they can significantly improve performance in a demand paging environment.

Question 5: Contributions

Keerthana Bidare (kbida003)

1. Implemented 4.1 and 4.2.1.
2. Implemented a part of 4.3.3.

Dhanush Radhakrishna (dradh003)

1. Implemented 4.3.1 and 4.3.2.
2. Implemented a part of 4.3.3.

Akshay Jayaram (ajaya026)

1. Completed the report.
2. Implemented 4.2.2 and helped in debugging.