CSO A2 Report

Task 0

System details were found out by running the following:

```
system_profiler SPSoftwareDataType SPHardwareDataType SPDisplaysDataType SPMemoryDataType SPNVMeDataType SPPCIDataType SPPowerDataType
sysctl -a | grep hw.l1 >> task0.txt
```

Software:

```
System Software Overview:

  System Version: macOS 11.2.3 (20D91)
  Kernel Version: Darwin 20.3.0
  Boot Volume: Macintosh HD
  Boot Mode: Normal
  Computer Name: Arihanth’s MacBook Pro
  User Name: Arihanth Tadanki (arihanthtadanki)
  Secure Virtual Memory: Enabled
  System Integrity Protection: Enabled
```

Hardware:

```
Hardware Overview:

  Model Name: MacBook Pro
  Model Identifier: MacBookPro16,1
  Processor Name: 6-Core Intel Core i7
  Processor Speed: 2.6 GHz
  Number of Processors: 1
  Total Number of Cores: 6
  L1 Cache: 32768 B
  L2 Cache (per Core): 256 KB
  L3 Cache: 12 MB
  Hyper-Threading Technology: Enabled
  Memory: 16 GB
  System Firmware Version: 1554.80.3.0.0 (iBridge: 18.16.14347.0.0,0)
  Serial Number (system): C02D779WMD6M
  Hardware UUID: 7503AEAE-8D83-5085-9F37-15880B86C012
  Provisioning UDID: 7503AEAE-8D83-5085-9F37-15880B86C012
  Activation Lock Status: Enabled
```

Graphics/Displays:

```
Intel UHD Graphics 630:

  Chipset Model: Intel UHD Graphics 630
  Type: GPU
  Bus: Built-In
  VRAM (Dynamic, Max): 1536 MB
  Vendor: Intel
  Device ID: 0x3e9b
  Revision ID: 0x0000
  Automatic Graphics Switching: Supported
  gMux Version: 5.0.0
  Metal Family: Supported, Metal GPUFamily macOS 2

AMD Radeon Pro 5300M:

  Chipset Model: AMD Radeon Pro 5300M
  Type: GPU
  Bus: PCIe
  PCIe Lane Width: x8
  VRAM (Total): 4 GB
  Vendor: AMD (0x1002)
  Device ID: 0x7340
  Revision ID: 0x0043
  ROM Revision: 113-D3220E-190
  VBIOS Version: 113-D32207P1-019
  Option ROM Version: 113-D32207P1-019
  EFI Driver Version: 01.A1.190
  Automatic Graphics Switching: Supported
```

```
  gMux Version: 5.0.0
  Metal Family: Supported, Metal GPUFamily macOS 2
  Displays:
    LG HDR 4K:
      Resolution: 5120 x 2880 (5K/UHD+ - Ultra High Definition Plus)
      UI Looks like: 2560 x 1440 @ 60.00Hz
      Framebuffer Depth: 30-Bit Color (ARGB2101010)
      Main Display: Yes
      Mirror: Off
      Online: Yes
      Rotation: Supported
      Adapter Type: DVI or HDMI
      Automatically Adjust Brightness: Yes
      Adapter Firmware Version: ff.c1
```

Memory:

```
Memory Slots:

  ECC: Disabled
  Upgradeable Memory: No

    BANK 0/ChannelA-DIMM0:

      Size: 8 GB
      Type: DDR4
      Speed: 2667 MHz
      Status: OK
      Manufacturer: Micron
      Part Number: 8ATF1G64HZ-2G6E1
      Serial Number: -

    BANK 2/ChannelB-DIMM0:

      Size: 8 GB
      Type: DDR4
      Speed: 2667 MHz
      Status: OK
      Manufacturer: Micron
      Part Number: 8ATF1G64HZ-2G6E1
      Serial Number: -
```

NVMExpress:

```
Apple SSD Controller:

    APPLE SSD AP0512N:

      Capacity: 500.28 GB (5,00,27,77,92,768 bytes)
      TRIM Support: Yes
      Model: APPLE SSD AP0512N
      Revision: 1161.80.
      Serial Number: C020326074ENFHT1E
      Link Width: x4
      Link Speed: 8.0 GT/s
      Detachable Drive: No
      BSD Name: disk0
      Partition Map Type: GPT (GUID Partition Table)
      Removable Media: No
      S.M.A.R.T. status: Verified
      Volumes:
        EFI:
          Capacity: 314.6 MB (31,45,72,800 bytes)
          File System: MS-DOS FAT32
          BSD Name: disk0s1
          Content: EFI
          Volume UUID: E783267B-A4C3-3556-B751-DBED770EB996
        disk0s2:
          Capacity: 499.96 GB (4,99,96,31,74,912 bytes)
          BSD Name: disk0s2
          Content: Apple_APFS
```

Power:

```
Battery Information:

  Model Information:
      Manufacturer: DSY
```

```
      Device Name: bq40z651
      Pack Lot Code: 0
      PCB Lot Code: 0
      Firmware Version: 0b00
      Hardware Revision: 300
      Cell Revision: 809
  Charge Information:
      Fully Charged: No
      Charging: No
      Full Charge Capacity (mAh): 7808
      State of Charge (%): 81
  Health Information:
      Cycle Count: 41
      Condition: Normal

System Power Settings:

  AC Power:
      System Sleep Timer (Minutes): 15
      Disk Sleep Timer (Minutes): 10
      Display Sleep Timer (Minutes): 15
      Wake on AC Change: No
      Wake on Clamshell Open: Yes
      Wake on LAN: Yes
      Current Power Source: Yes
      Display Sleep Uses Dim: Yes
      Hibernate Mode: 3
      PrioritizeNetworkReachabilityOverSleep: 0
  Battery Power:
      System Sleep Timer (Minutes): 1
      Disk Sleep Timer (Minutes): 10
      Display Sleep Timer (Minutes): 15
      Wake on AC Change: No
      Wake on Clamshell Open: Yes
      Display Sleep Uses Dim: Yes
      Hibernate Mode: 3
      Reduce Brightness: No

Hardware Configuration:

  UPS Installed: No

AC Charger Information:

  Connected: Yes
  ID: 0x7002
  Wattage (W): 94
  Family: 0xe000400a
  Serial Number: C4H029408BTL4YTAF
  Name: 96W USB-C Power Adapter
  Manufacturer: Apple Inc.
  Hardware Version: 1.0
  Firmware Version: 1070051
  Charging: No

Power Events:

    Next Scheduled Events:

          appPID: 382
          Type: Wake
          Scheduled By: com.apple.alarm.user-visible-Weekly Usage Report
          Time: 25/04/21, 3:17 AM
          UserVisible: 0

          appPID: 382
          Type: Wake
          Scheduled By: com.apple.alarm.user-visible-Weekly Usage Report
          Time: 25/04/21, 5:17 AM
          UserVisible: 0
```

Thunderbolt/USB4:

```
Thunderbolt Bus 1:

  Vendor Name: Apple Inc.
  Device Name: MacBook Pro
  UID: 0x0001F40FAC51E901
  Route String: 0
  Firmware Version: 63.5
  Domain UUID: 6670C20A-C4D7-C755-8977-6B9EE2FBAB42
  Port:
      Status: No device connected
      Link Status: 0x101
```

```
      Speed: Up to 40 Gb/s x1
      Current Link Width: 0x1
      Receptacle: 4
      Link Controller Firmware Version: 1.43.0
  Port:
      Status: No device connected
      Link Status: 0x101
      Speed: Up to 40 Gb/s x1
      Current Link Width: 0x1
      Receptacle: 3
      Link Controller Firmware Version: 1.43.0

Thunderbolt Bus 0:

  Vendor Name: Apple Inc.
  Device Name: MacBook Pro
  UID: 0x0001F40FAC51E900
  Route String: 0
  Firmware Version: 63.5
  Domain UUID: A9FE930F-C6D7-2A50-9D12-9DFD9AE12413
  Port:
      Status: No device connected
      Link Status: 0x101
      Speed: Up to 40 Gb/s x1
      Current Link Width: 0x1
      Receptacle: 2
      Link Controller Firmware Version: 1.43.0
  Port:
      Status: No device connected
      Link Status: 0x101
      Speed: Up to 40 Gb/s x1
      Current Link Width: 0x1
      Receptacle: 1
      Link Controller Firmware Version: 1.43.0
```

USB:

```
USB 3.1 Bus:

  Host Controller Driver: AppleUSBXHCITR
  PCI Device ID: 0x15ec
  PCI Revision ID: 0x0006
  PCI Vendor ID: 0x8086
  Bus Number: 0x01

USB 3.1 Bus:

  Host Controller Driver: AppleUSBXHCITR
  PCI Device ID: 0x15ec
  PCI Revision ID: 0x0006
  PCI Vendor ID: 0x8086
  Bus Number: 0x00

    USB3.1 Hub:

      Product ID: 0x1018
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 42.06
      Serial Number: DLC928606SZKD5664
      Speed: Up to 5 Gb/s
      Manufacturer: Apple Inc.
      Location ID: 0x00200000 / 1
      Current Available (mA): 900
      Current Required (mA): 0
      Extra Operating Current (mA): 0

USB 3.1 Bus:

  Host Controller Driver: AppleIntelCNLUSBXHCI
  PCI Device ID: 0xa36d
  PCI Revision ID: 0x0010
  PCI Vendor ID: 0x8086

    USB2.0 Hub:

      Product ID: 0x1017
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 42.06
      Serial Number: DLC928606SZKD5664
      Speed: Up to 480 Mb/s
      Manufacturer: Apple Inc.
      Location ID: 0x14400000 / 1
      Current Available (mA): 500
      Current Required (mA): 0
```

```
      Extra Operating Current (mA): 0

        USB-C Digital AV Multiport Adapter:

          Product ID: 0x1463
          Vendor ID: 0x05ac (Apple Inc.)
          Version: 42.06
          Serial Number: DLC928606SZKD5664
          Speed: Up to 12 Mb/s
          Manufacturer: Apple Inc
          Location ID: 0x14420000 / 2
          Current Available (mA): 500
          Current Required (mA): 100
          Extra Operating Current (mA): 0

Apple T2 Bus:

  Host Controller Driver: AppleUSBVHCIBCE

    Composite Device:

      Product ID: 0x8104
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.14
      Serial Number: 000000000000
      Speed: Up to 480 Mb/s
      Manufacturer: Apple
      Location ID: 0x80800000 / 4
      Current Available (mA): 500
      Current Required (mA): 500
      Extra Operating Current (mA): 0
      Built-In: Yes

    Touch Bar Backlight:

      Product ID: 0x8102
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.01
      Serial Number: 0000000000000000
      Manufacturer: Apple Inc.
      Location ID: 0x80700000

    Touch Bar Display:

      Product ID: 0x8302
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.01
      Serial Number: 0000000000000000
      Manufacturer: Apple Inc.
      Location ID: 0x80600000

    Apple Internal Keyboard / Trackpad:

      Product ID: 0x0340
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.75
      Serial Number: FM7032103CPHYYKBL+TRZ
      Manufacturer: Apple Inc.
      Location ID: 0x80500000

    Headset:

      Product ID: 0x8103
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.14
      Serial Number: 000000000000
      Manufacturer: Apple
      Location ID: 0x80400000

    Ambient Light Sensor:

      Product ID: 0x8262
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.01
      Serial Number: 000000000000
      Manufacturer: Apple Inc.
      Location ID: 0x80300000

    FaceTime HD Camera (Built-in):

      Product ID: 0x8514
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.01
      Serial Number: CC2027603Z2HNW1AX
      Manufacturer: Apple Inc.
      Location ID: 0x80200000
```

```
    Apple T2 Controller:

      Product ID: 0x8233
      Vendor ID: 0x05ac (Apple Inc.)
      Version: 2.01
      Serial Number: 0000000000000000
      Manufacturer: Apple Inc.
      Location ID: 0x80100000
```

Kernel Modules:

```
Index Refs Address            Size       Wired      Name (Version) UUID <Linked Against>
1  143 0                  0          0          com.apple.kpi.bsd (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
2   11 0                  0          0          com.apple.kpi.dsep (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
3  178 0                  0          0          com.apple.kpi.iokit (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
4    0 0                  0          0          com.apple.kpi.kasan (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
5  181 0                  0          0          com.apple.kpi.libkern (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
6  159 0                  0          0          com.apple.kpi.mach (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
7   93 0                  0          0          com.apple.kpi.private (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
8  107 0                  0          0          com.apple.kpi.unsupported (20.3.0) C86236B2-4976-3542-80CA-74A6B8B4BA03 <>
9    2 0xffffff80031e8000 0xe000     0xe000     com.apple.kec.Libm (1) 75E2671F-F050-3765-96B8-BD03232E1A62 <5>
10   12 0xffffff8003406000 0x91000    0x91000    com.apple.kec.corecrypto (11.1) E009770B-BAFB-3BF9-B176-D6B4CC96DF5F <8 7 6 5 3 1>
11    0 0xffffff80034a1000 0x8000     0x8000     com.apple.kec.pthread (1) 1E922D2A-9975-3AFB-A009-73EC3EF87A86 <8 7 6 5 3 1>
12    1 0xffffff8003563000 0x2000     0x2000     com.apple.driver.watchdog (1) 2C1307C5-FD71-3802-8D68-A67DB24BCB4A <8 7 6 5 3 1>
13   41 0xffffff80026d8000 0x2000     0x2000     com.apple.iokit.IOACPIFamily (1.4) B41AA072-8BB9-3493-9796-6FF849CDABB8 <8 7 5 3>
14   48 0xffffff8002c1f000 0x28000    0x28000    com.apple.iokit.IOPCIFamily (2.9) BF2C5E86-1E8F-3FD4-9874-7738178FA73B <8 7 6 5 3>
15   14 0xffffff8001895000 0x19000    0x19000    com.apple.driver.AppleSMC (3.1.9) 1E48B8EB-7BF3-3366-8520-2C4B31A7D3F8 <14 13 12 8 7
16    2 0xffffff8001292000 0x77000    0x77000    com.apple.driver.AppleACPIPlatform (6.1) 2F2AA264-6BD5-3E4F-829C-8952B3ABB2D5 <15 14
17    5 0xffffff8002d37000 0x2000     0x2000     com.apple.driver.IOSlaveProcessor (1) FC8BCD16-0792-35FD-B23E-A0720BC40E1A <5 3>
18    3 0xffffff800187b000 0x10000    0x10000    com.apple.driver.AppleSEPManager (1.0.1) 4FC2AB67-A77F-3028-BFF6-7BE23EEE6773 <17 14
19    1 0xffffff800158c000 0x4000     0x4000     com.apple.driver.AppleBusPowerController (1.0) 499DB0FD-1FE4-31FC-B0E0-ECE24BF29417 <
20   14 0xffffff80019e2000 0x4000     0x4000     com.apple.driver.usb.AppleUSBCommon (1.0) 62FE0E66-DCCA-3443-B01E-0F0FAB0A820D <6 5 3
21    4 0xffffff800304d000 0x1000     0x1000     com.apple.driver.AppleUSBHostMergeProperties (1.2) 90FEE882-3237-3822-91F1-6AD62C8E46
22   24 0xffffff8002f52000 0x8d000    0x8d000    com.apple.iokit.IOUSBHostFamily (1.2) A84D73D5-4F62-32E2-A1DB-9F021795AE81 <21 20 19
23    3 0xffffff80031d2000 0xf000     0xf000     com.apple.driver.KernelRelayHost (1) 6E0EDF05-F095-3009-AC2B-BE4CEFFF6583 <22 6 5 3>
24    1 0xffffff8001600000 0x50000    0x50000    com.apple.driver.AppleCredentialManager (1.0) 56AFE766-6003-3409-B493-402505841C1C <2
25    3 0xffffff800166d000 0x6000     0x6000     com.apple.driver.AppleEffaceableStorage (1.0) CE527E76-F27F-33BD-933A-ED837F559A6D <8
26    0 0xffffff8001677000 0x6000     0x6000     com.apple.driver.AppleFDEKeyStore (28.30) FDFE34C9-00DA-3C05-AC10-14CF0EB24902 <25 10
27    2 0xffffff8001b7e000 0x7000     0x7000     com.apple.kext.CoreTrust (1) 36944A7C-F451-35A4-82A1-3B6BF8CBAC71 <10 5>
28    9 0xffffff8001801000 0x15000    0x15000    com.apple.driver.AppleMobileFileIntegrity (1.0.5) 2A454117-CDAA-301F-B609-BA396742C91
29    3 0xffffff8002c5a000 0x14000    0x14000    com.apple.iokit.IOSCSIArchitectureModelFamily (436.40.6) E5CB0567-FBD1-39AE-8847-C559
30   10 0xffffff8002d3f000 0x12000    0x12000    com.apple.iokit.IOStorageFamily (2.1) B5300908-BF34-3D47-8776-FB154A6DEE4C <8 7 6 5 3
31    1 0xffffff8002c7f000 0xe000     0xe000     com.apple.iokit.IOSCSIBlockCommandsDevice (436.40.6) 17662AB7-6B15-3342-9F23-AB161202
32    1 0xffffff80031ae000 0x1a000    0x1a000    com.apple.iokit.IOUSBMassStorageDriver (184.40.6) AB4AC255-99A6-38E2-B5FE-22B97EEC79A
33    2 0xffffff80018ef000 0x8000     0x8000     com.apple.driver.AppleUSBTDM (511.60.2) E381525B-81E2-3E87-B197-6355B228751B <32 31 3
34    0 0xffffff800175c000 0x80000    0x80000    com.apple.driver.AppleKeyStore (2) EAF92490-F231-30E8-A019-2C01F28E4315 <33 28 27 24
35    0 0xffffff80018b6000 0x7000     0x7000     com.apple.driver.AppleSSE (1.0) 59F98016-B7A6-3538-8C5E-ABF400FA3EE6 <23 18 17 8 6 5
36    4 0xffffff8001aa9000 0x7000     0x7000     com.apple.iokit.CoreAnalyticsFamily (1) CB8BC944-2B59-355C-A5D6-05FE908EBF67 <8 7 6 5
37    2 0xffffff80017fd000 0x3000     0x3000     com.apple.kext.AppleMatch (1.0.0d1) 82D16ED7-54AB-3562-9ABF-05E96C622BF6 <5 1>
38    3 0xffffff800321a000 0x44000    0x44000    com.apple.security.sandbox (300.0) 2BC91DC8-F0AF-3762-B99A-53A1CF504978 <37 30 28 8 7
39    2 0xffffff8003211000 0x8000     0x8000     com.apple.security.quarantine (4) 1C99CAC8-F001-3CEE-92E1-BB2CFC5A4D8F <38 37 8 7 6 5
40    0 0xffffff8001b86000 0x4c000    0x4c000    com.apple.iokit.EndpointSecurity (1) 77BEC648-BF3B-3BC3-B2D9-27CA8967CC51 <39 28 8 7
41    1 0xffffff8002858000 0x6000     0x6000     com.apple.iokit.IOBluetoothPacketLogger (8.0.3d9) 097BD1D7-B43F-3B74-B331-C40C51F32A4
42   11 0xffffff8002c55000 0x3000     0x3000     com.apple.iokit.IOReportFamily (47) D3C4FAA4-8F06-3C5C-AB36-4BE632CCE051 <6 5 3>
43    4 0xffffff800276a000 0x9e000    0x9e000    com.apple.iokit.IOBluetoothFamily (8.0.3d9) EF7A0345-A992-34DD-A26F-C853D623D78B <42
44    0 0xffffff8002895000 0xc000     0xc000     com.apple.driver.DiskImages (493.0.0) 11E51AF9-6467-39AC-89A2-A62CEF763F298 <30 8 7 6
45    0 0xffffff8002962000 0x1000     0x1000     com.apple.iokit.IOKitRegistryCompatibility (1) 2D92E372-8280-3FC3-B646-9FB3667B8D3D <
46   13 0xffffff8002996000 0x17000    0x17000    com.apple.iokit.IONetworkingFamily (3.4) CEA28E53-3048-362D-B58E-3A02FFC20EA9 <8 7 6
47    3 0xffffff8002e87000 0x1b000    0x1b000    com.apple.iokit.IOTimeSyncFamily (900.11) E181C003-3A97-339B-BD7D-26FECFDA8CF2 <46 6
48    0 0xffffff8001288000 0x9000     0x9000     com.apple.nke.applicationfirewall (310) D31A7A1B-3D95-3727-BC98-719DD7756B8D <8 7 6 5
49    1 0xffffff80016cf000 0x10000    0x10000    com.apple.security.AppleImage4 (3.0.0) 5D746F35-15A8-3406-8725-8867038D656D <10 8 7 5
50    0 0xffffff800190b000 0xe000     0xe000     com.apple.AppleSystemPolicy (2.0.0) 6756BBFE-4F94-3A32-AF9A-377AE3136B0E <39 38 28 8
51    0 0xffffff800132f000 0x2000     0x2000     com.apple.driver.AppleAPIC (1.7) A598BD53-CDC1-3CDB-B9A3-5D504B859FA9 <14 5 3>
52   11 0xffffff80028d2000 0x5d000    0x5d000    com.apple.iokit.IOHIDFamily (2.0.0) 4A62A5E5-1CB2-3B05-8BBD-5044EF737AC0 <42 8 7 6 5
53    3 0xffffff8002cd8000 0x1000     0x1000     com.apple.iokit.IOSMBusFamily (1.1) 1B91C74F-E16F-30B0-93C2-B16EA181DC59 <6 5 3>
54    0 0xffffff800131c000 0x5000     0x5000     com.apple.driver.AppleACPIEC (6.1) 0912BC2E-5CBA-3EA6-9B82-59C837DC754B <53 52 42 16
55    0 0xffffff8001891000 0x3000     0x3000     com.apple.driver.AppleSMBIOS (2.1) 6AD3FBAC-858E-3372-8DAF-69F54979BF82 <8 5 3>
56    0 0xffffff80018b1000 0x4000     0x4000     com.apple.driver.AppleSMCRTC (1.0) A8B4C6FC-A4D8-344F-9E62-DE2EDC85758F <15 13 8 6 5
57    0 0xffffff8001317000 0x2000     0x2000     com.apple.driver.AppleACPIButtons (6.1) 0CFFA69C-CBF7-3998-9677-9937E377E15D <52 42 1
58    1 0xffffff8001657000 0x2000     0x2000     com.apple.driver.AppleEFIRuntime (2.1) DFD17EE5-E4FE-319E-9191-EFEEA53690DA <8 7 6 5
59    2 0xffffff800165a000 0xa000     0xa000     com.apple.driver.AppleEFINVRAM (2.1) D6C13E44-3657-3F40-99E4-355DAA82202E <58 8 7 6 5
60    0 0xffffff80018bf000 0x11000    0x11000    com.apple.driver.AppleSmartBatteryManager (161.0.0) 81E73A5A-759D-3F2D-923D-0803C9F7A
61    0 0xffffff80031e5000 0x1000     0x1000     com.apple.private.KextAudit (1.0) F8D19031-5AF3-30AC-982F-D51A7A941DBD <15 8 7 6 5 3>
62    1 0xffffff80030f0000 0x4f000    0x4f000    com.apple.driver.usb.AppleUSBXHCI (1.2) D623BBDD-59E1-3494-BE57-0F4AFD1CF853 <22 20 1
63    0 0xffffff8003148000 0x28000    0x28000    com.apple.driver.usb.AppleUSBXHCIPCI (1.2) 279DC1E0-EDF4-3931-83DC-CF11DC249E98 <62 2
64    0 0xffffff8002eaf000 0x63000    0x63000    com.apple.iokit.IOUSBFamily (900.4.2) F474B853-D7A0-3C87-BAD8-8C16049A08F8 <22 20 14
65    0 0xffffff800304f000 0x2000     0x2000     com.apple.driver.usb.AppleUSBHostPacketFilter (1.0) 6D208E1B-AE2C-3DE4-80DE-2B1D0F5C4
66    8 0xffffff8002d87000 0xd9000    0xd9000    com.apple.iokit.IOThunderboltFamily (9.3.2) 73A0BDC8-85DE-3B44-99A7-A864CA44D289 <15
67    0 0xffffff8001964000 0x30000    0x30000    com.apple.driver.AppleThunderboltNHI (7.2.8) CD53D1A8-1CA7-3041-BC61-EF4A58B0758B <66
68    0 0xffffff8002964000 0x2a000    0x2a000    com.apple.iokit.IONVMeFamily (2.1.0) D5DFC80E-EF7A-3660-BE57-473E67626B44 <59 42 30 2
69    6 0xffffff800287b000 0x7000     0x7000     com.apple.iokit.IOBufferCopyEngineFamily (1) D15BDB5B-EA3F-38AB-86DA-F8E277140376 <6
70    0 0xffffff8002875000 0x4000     0x4000     com.apple.iokit.IOBufferCopyController (1.1.0) 7E14DE49-716A-3CB6-A5C8-636707EF8328 <
71    0 0xffffff8001666000 0x4000     0x4000     com.apple.driver.AppleEffaceableNOR (1.0) 3C97037C-CFE2-3EAC-A629-BF690A3F61E1 <69 33
72    3 0xffffff80019e7000 0x6000     0x6000     com.apple.driver.usb.AppleUSBVHCICommon (1.0) AF314B32-BF81-3386-9D5D-9CBE97F38A48 <2
```

```
 73    1 0xffffff80030cb000 0x1e000    0x1e000    com.apple.driver.usb.AppleUSBVHCI (1.2) 630BF562-58F3-38DE-B1D7-C96F93B822B4 <72 22 2
 74    1 0xffffff80019ee000 0xb000     0xb000     com.apple.driver.usb.AppleUSBVHCICommonBCE (1.0) 556E19FE-3DA5-3780-A5C9-CB649031C9B8
 75    0 0xffffff80030ee000 0x1000     0x1000     com.apple.driver.usb.AppleUSBVHCIBCE (1.2) C28DAF11-2F78-319A-BCEC-82463267126A <74 7
 76    0 0xffffff800167f000 0x2000     0x2000     com.apple.AppleFSCompression.AppleFSCompressionTypeDataless (1.0.0d1) 1C6A2FF2-53DF-3
 77    0 0xffffff8001682000 0x9000     0x9000     com.apple.AppleFSCompression.AppleFSCompressionTypeZlib (1.0.0) F986F845-36B3-34A6-86
 78    0 0xffffff8001a9b000 0xc000     0xc000     com.apple.BootCache (40) 5FF5DE96-C153-37AE-96BA-057BB841EFD9 <8 7 6 5 3 1>
 79    1 0xffffff8001c3d000 0x2000     0x2000     com.apple.filesystems.hfs.encodings.kext (1) 99860746-6E6D-383E-BBAD-9EA92F3C189F <8
 80    0 0xffffff8001bd8000 0x63000    0x63000    com.apple.filesystems.hfs.kext (556.60.1) CA235EAE-E294-3F07-B75B-36CF5D5287E4 <79 8
 81    2 0xffffff8002c19000 0x5000     0x5000     com.apple.driver.mDNSOffloadUserClient (1.0.1b8) 35ABC470-4743-30AC-9BA8-4A62CFA2296D
 82    5 0xffffff8002ce9000 0x3e000    0x3e000    com.apple.iokit.IOSkywalkFamily (1) 882D8199-A1D6-3A53-9A83-1D11A15DECFA <81 46 8 7 6
 83    3 0xffffff8002d66000 0x1c000    0x1c000    com.apple.iokit.IOSurface (289.3) 22DEF369-CE8F-3ACF-BD1E-4CE16AAB209A <8 7 6 5 3 1>
 84    0 0xffffff8001567000 0x1000     0x1000     com.apple.driver.AppleBSDKextStarter (3) F512DB7A-E949-364B-893A-B155C919EF2F <5 3>
 85    0 0xffffff80034aa000 0xc000     0xc000     com.apple.filesystems.tmpfs (1) 33EBC2A2-A8BD-33B8-B821-0FB4F5B2EC77 <8 7 6 5 1>
 86    0 0xffffff800168e000 0x1000     0x1000     com.apple.driver.AppleFileSystemDriver (3.0.1) C8E4EE9E-6F90-3567-82F3-AEAE644B74D1 <
 87    0 0xffffff8001a67000 0x3000     0x3000     com.apple.driver.AppleXsanScheme (3) B44336C1-BE52-391B-AEF3-46B930C214EA <30 6 5 3 1
 88    1 0xffffff8003276000 0x163000   0x163000   com.apple.filesystems.apfs (1677.81.1) DB8F6842-2D8F-3535-8352-54EBDD26900F <59 49 36
 89    1 0xffffff8001724000 0x5000     0x5000     com.apple.driver.AppleIntelLpssDmac (3.0.60) 9959CA36-3989-3C3F-8DEF-13CD5D483061 <14
 90    1 0xffffff8001733000 0x7000     0x7000     com.apple.driver.AppleIntelLpssI2C (3.0.60) 90CCC45D-CADA-3BF5-9F32-FF2394206605 <14
 91    1 0xffffff800173b000 0xd000     0xd000     com.apple.driver.AppleIntelLpssI2CController (3.0.60) 2B34E51B-1C82-331B-A9CF-EC0E7EC
 92    0 0xffffff80016a5000 0x6000     0x6000     com.apple.driver.AppleHPM (3.4.4) CE3C1B8D-EBE9-33E4-B8DE-966FD768613A <91 66 13 6 5
 93    3 0xffffff800191c000 0x17000    0x17000    com.apple.driver.AppleThunderboltDPAdapterFamily (8.1.4) 83B22C55-DF42-30CF-89B2-73EE
 94    0 0xffffff8001935000 0xa000     0xa000     com.apple.driver.AppleThunderboltDPInAdapter (8.1.4) 7D379694-E68C-3A06-8258-5FD90C30
 95    0 0xffffff800199d000 0x4000     0x4000     com.apple.driver.AppleThunderboltPCIDownAdapter (4.1.1) CDEFA2DE-9DEF-3B64-89A9-737BA
 96    8 0xffffff80033e4000 0x17000    0x17000    com.apple.driver.corecapture (1.0.4) DDCF23B4-962A-3A18-B089-CD07BDAD9A35 <8 7 6 5 3
 97    6 0xffffff800294d000 0x11000    0x11000    com.apple.driver.IOImageLoader (1.0.0) 4430E4D9-A9A6-33B5-B111-F7576916BF73 <96 10 8
 98    2 0xffffff8002580000 0x13e000   0x13e000   com.apple.iokit.IO80211FamilyV2 (1200.12.2b1) FA0689F9-1CFE-351D-8B83-117E0BEEB442 <9
 99    6 0xffffff8002cda000 0x6000     0x6000     com.apple.iokit.IOSerialFamily (11) BFD092AF-19A4-37DD-AD69-6C36365D8044 <8 7 6 5 3 1
100    1 0xffffff8001395000 0x1b5000   0x1b5000   com.apple.driver.AppleBCMWLANCoreMac (1.0.0) 45916170-3DA2-3E28-8A23-5302B00F90E6 <9
101    0 0xffffff800133c000 0x4f000    0x4f000    com.apple.driver.AppleBCMWLANBusInterfacePCIeMac (1) A1B93693-3023-3956-9BE3-CA8E13A
102    0 0xffffff8001a7e000 0x2000     0x2000     com.apple.driver.BCMWLANFirmware4355.Hashstore (1) 7BE8814D-5477-331F-85A9-82D1F63EA
103    0 0xffffff8001a81000 0x7000     0x7000     com.apple.driver.BCMWLANFirmware4364.Hashstore (1) 6A851A21-C71E-3DA2-A622-1E6B23D7C
104    0 0xffffff8001a8d000 0x4000     0x4000     com.apple.driver.BCMWLANFirmware4377.Hashstore (1) 74B07645-E4EC-311F-9C5C-5F7490134
105    0 0xffffff8001a94000 0x4000     0x4000     com.apple.driver.BCMWLANFirmware4378.Hashstore (1) CF530FF8-7947-36EE-8A37-3D69E0878
106    3 0xffffff8003048000 0x3000     0x3000     com.apple.driver.usb.AppleUSBHostCompositeDevice (1.2) E262EB63-C5DE-39E7-81A8-7D392
107    3 0xffffff8001a2a000 0x2000     0x2000     com.apple.driver.usb.networking (5.0.0) 9B5DD9DF-9A0D-31C5-80E0-670D6AFD6EF5 <22 7 6
108    2 0xffffff80019de000 0x2000     0x2000     com.apple.driver.usb.cdc (5.0.0) B246DA0A-7913-37F2-AB4E-38CD5B154A19 <107 106 22 6
109    0 0xffffff8003052000 0x2b000    0x2b000    com.apple.driver.usb.AppleUSBHub (1.2) 7699C6C9-BEF3-358D-817A-C2F1C8D0DEDD <22 20 1
110    0 0xffffff8001a21000 0x7000     0x7000     com.apple.driver.usb.cdc.ncm (5.0.0) CD9A1845-EEE9-39FF-A333-D36D98037AEB <108 107 1
111    0 0xffffff80019fd000 0x3000     0x3000     com.apple.driver.usb.cdc.ecm (5.0.0) 99DA5868-03A9-3B5B-8B88-DFA804972D46 <108 107 4
112    0 0xffffff80031a2000 0xa000     0xa000     com.apple.driver.usb.IOUSBHostHIDDevice (1.2) 76E6A984-09DA-3A1D-93F8-8975E8AC06DA <
113    0 0xffffff8003045000 0x2000     0x2000     com.apple.driver.usb.AppleUSBHostBillboardDevice (1.0) 83258EC3-7FF2-36E4-8979-A6849
114    1 0xffffff7fa0684000 0x7000     0x7000     com.apple.driver.AppleSMBusController (1.0.18d1) D69C0758-0371-369E-88ED-E11402B9100
115   21 0xffffff7fa0a63000 0x2f000    0x2f000    com.apple.iokit.IOGraphicsFamily (585) EF8876FC-C6D6-3D53-9915-762EF4F12870 <14 8 6
116    0 0xffffff7fa0634000 0xb000     0xb000     com.apple.driver.AppleMCCSControl (1.14) 4D16F699-71E8-3335-A706-A534F1D2B993 <115 1
117    4 0xffffff7fa0aa8000 0x9000     0x9000     com.apple.iokit.IONDRVSupport (585) 40BEBEA2-BBB5-3B09-B6AE-B4B372775FBD <115 14 8 6
118    2 0xffffff7f9efe8000 0x2000     0x2000     com.apple.driver.AppleBacklightExpert (1.1.0) 29E038A5-5B0C-37EA-A92C-52C8744FF59F <
119    0 0xffffff7f9efe3000 0x3000     0x3000     com.apple.driver.AppleBacklight (180.3) 13F78CBD-7588-3F9A-893A-50DE0B154B96 <118 11
120    9 0xffffff7f9f1b7000 0x4000     0x4000     com.apple.AppleGraphicsDeviceControl (6.2.2) 867CDCFC-72A3-390B-B252-5F0F9390C1AF <1
121    0 0xffffff7f9f1a5000 0x3000     0x3000     com.apple.driver.AGDCBacklightControl (6.2.2) A4C0D46D-1AD5-3CE6-B9D5-F12AA9AFCBF8 <
122    0 0xffffff7f9f0bb000 0x2000     0x2000     com.apple.driver.AppleFIVRDriver (4.1.0) E9BC9DF9-6CA6-3AB5-9EBD-21C6D8FDC814 <5 3>
123    6 0xffffff7fa0ab3000 0x4000     0x4000     com.apple.driver.IOPlatformPluginFamily (6.0.0d8) 781FB738-E551-3048-A081-CDA1E20922
124    0 0xffffff7fa0302000 0x3000     0x3000     com.apple.driver.AppleIntelPCHPMC (2.0.1) 81631993-0DC4-315E-A630-6F46AA1C4699 <123
125    3 0xffffff7fa090b000 0x6a000    0x6a000    com.apple.iokit.IOAcceleratorFamily2 (439.52) 7B467C8D-2528-3663-8024-B09E1DCED75A <
126    0 0xffffff7f9f653000 0xbf000    0xbf000    com.apple.driver.AppleIntelCFLGraphicsFramebuffer (16.0.1) 9E24E6AB-A78F-3DF2-B7FA-2
128    2 0xffffff7fa0646000 0x10000    0x10000    com.apple.driver.AppleOnboardSerial (1.0) E6BCA53E-2D47-3C3C-8197-59E1F2F8CEB1 <99 8
129    1 0xffffff7fa02ea000 0x1000     0x1000     com.apple.driver.AppleIntelLpssUARTCommon (3.0.60) 2ADB5B9B-4528-3711-91BF-5D75CD70A
130    0 0xffffff7fa02ee000 0x1000     0x1000     com.apple.driver.AppleIntelLpssUARTv1 (3.0.60) B274A573-702A-3489-AEA4-C867ECC82BE2
131    0 0xffffff7fa068f000 0x1000     0x1000     com.apple.driver.AppleSMBusPCI (1.0.14d1) 08A4B6D9-54C6-3632-B72C-69FA0625A0AB <14 6
132    1 0xffffff80034b7000 0xab000    0xab000    com.apple.vecLib.kext (1.2.0) 528BC21D-5AEB-365D-98CD-E3489B4C37DA <9 8 7 6 5 3>
133    4 0xffffff800273b000 0x1e000    0x1e000    com.apple.iokit.IOAudioFamily (300.6.1) 8CBD98F6-7011-35DA-B0A7-47EF6636B572 <132 6
134    0 0xffffff7f9f12d000 0x5e000    0x5e000    com.apple.driver.AppleGFXHDA (100.1.431) 71E02A6A-3166-397B-B393-85C77BDCC615 <133 1
135    1 0xffffff7f9f405000 0x4000     0x4000     com.apple.iokit.IOHDAFamily (283.15) CFE5D674-E1F8-3EFC-8DEB-50FFF11C8D57 <6 5 3 1>
136    0 0xffffff7f9f2e5000 0x13000    0x13000    com.apple.driver.AppleHDAController (283.15) 2180888A-8340-3930-9534-C3693D9599EA <1
137    0 0xffffff7f9efd2000 0x3000     0x3000     com.apple.driver.AppleAVEBridge (6.1) CAEE82E8-2157-3454-8336-52FB22DA106A <69 6 5 3
138    0 0xffffff7fa078a000 0xa000     0xa000     com.apple.driver.BridgeAudioCommunication (100.2) 6AF75392-B4C3-371D-8554-ACF022FCD9
139    1 0xffffff8002d3c000 0x2000     0x2000     com.apple.iokit.IOSlowAdaptiveClockingFamily (1.0.0) A0FE30D2-102E-334B-A1CD-660522D
140    0 0xffffff7fa062f000 0x1000     0x1000     com.apple.driver.AppleIntelSlowAdaptiveClocking (4.0.0) E418234E-462F-3CC8-A573-1707
141    0 0xffffff7f9f1ad000 0x7000     0x7000     com.apple.AppleGPUWrangler (6.2.2) 6F5CDA38-68B4-301D-B459-CEC6F7186D0F <120 115 14
142    0 0xffffff7f9f0a2000 0xc000     0xc000     com.apple.driver.AppleDiskImages2 (1) E6E4351B-D4E8-3F24-AE7D-45B665E4C4E5 <30 6 5 3
143    1 0xffffff8001941000 0x20000    0x20000    com.apple.driver.AppleThunderboltDPOutAdapter (8.1.4) 7CCEE4F8-910A-358B-AB50-92DCE1
144    0 0xffffff7fa06ba000 0x8000     0x8000     com.apple.driver.AppleThunderboltEDMSink (5.0.3) 4CC670D9-7A68-3B03-B240-CBA51D27C9C
145    0 0xffffff7fa06cb000 0x3d000    0x3d000    com.apple.driver.AppleThunderboltIP (4.0.3) 8F94B103-956C-3E84-B0A7-176139F7DC0C <66
146    1 0xffffff7fa0ada000 0x14000    0x14000    com.apple.driver.X86PlatformPlugin (1.0.0) 0A781B73-564E-3AA8-AFF9-718BCF2B9712 <123
147    1 0xffffff8002833000 0x5000     0x5000     com.apple.iokit.IOBluetoothHostControllerTransport (8.0.3d9) A5E9BD21-89A1-30A6-85C2
148    0 0xffffff8002839000 0x4000     0x4000     com.apple.iokit.IOBluetoothHostControllerUARTTransport (8.0.3d9) 7BC53107-D780-3C47-
149    2 0xffffff7f9f1a3000 0x1000     0x1000     com.apple.driver.AppleGraphicsControl (6.2.2) B1CF4471-53CC-316E-BFB2-1C94052E6ED0 <
150    0 0xffffff7f9f1f3000 0x13000    0x13000    com.apple.driver.AppleMuxControl2 (6.2.2) 52AF6D05-1C5A-3084-85C1-2379EF12ED2A <149
151    0 0xffffff7f9ffc2000 0xb8000    0xb8000    com.apple.driver.AppleIntelKBLGraphics (16.0.1) 9F686C76-339E-3B4C-91F9-B9A742F8B1CB
152    2 0xffffff8002c05000 0x2000     0x2000     com.apple.iokit.IOEthernetAVBController (1.1.0) CE70C64A-A310-3F71-B061-88A5887A8941
153    1 0xffffff7fa0b0e000 0x68000    0x68000    com.apple.plugin.IOgPTPPlugin (900.11) F295D470-2D07-3050-873C-0F7948844A95 <152 82
154    0 0xffffff7fa0797000 0x9000     0x9000     com.apple.driver.AppleBridgeAudioController (100.2) 2F7B3FA2-06CB-32F5-B9EB-E23462B2
155    0 0xffffff7f9f40d000 0xa000     0xa000     com.apple.driver.AppleHV (1) E718C011-691A-35C1-B0A7-AEC1025E19CA <38 8 7 6 5 3 1>
156    0 0xffffff7fa07a3000 0x3000     0x3000     com.apple.Dont_Steal_Mac_OS_X (7.0.0) C23783B5-21C5-3084-B67A-BB994F502C6A <15 10 8
157    1 0xffffff7fa0acc000 0x9000     0x9000     com.apple.driver.IOPlatformPluginLegacy (1.0.0) 7A544E94-68F4-3F68-9B91-09614A6ECFC3
158    0 0xffffff7fa0ab9000 0xd000     0xd000     com.apple.driver.ACPI_SMC_PlatformPlugin (1.0.0) 1CF1AC58-D4EC-3AC6-93EE-FE98EC533C4
159    0 0xffffff7f93894000 0x3000     0x3000     com.apple.kext.AMDRadeonX6000HWServices (4.0.2) FC3B89E9-C339-35BC-B688-09A8D8AF41F8
160    0 0xffffff7fa08a6000 0x13000    0x13000    com.apple.iokit.IOAVBFamily (930.1) 6C5052C9-A983-3626-863C-171013064822 <153 152 47
```

```
161    0 0xffffff7f9ec7d000 0x11a000   0x11a000   com.apple.kext.AMDSupport (4.0.2) 1F344D10-5EEE-3A67-9541-C7CACA8E84F3 <120 115 14 1
162    0 0xffffff7f9356b000 0x26f000   0x26f000   com.apple.kext.AMDRadeonX6000Framebuffer (4.0.2) EF0AE92F-0FA8-35D2-9ADD-57A98B3E003
163    0 0xffffff7fa0994000 0x7000     0x7000     com.apple.iokit.IOBluetoothSerialManager (8.0.3d9) B780E5DD-B4DD-3355-A631-9CFEAF0CF
164    0 0xffffff7fa077a000 0x3000     0x3000     com.apple.driver.AppleUpstreamUserClient (3.6.8) 1F766D0B-7CF1-3D05-AF80-2A7D62003F7
165    0 0xffffff7fa0b9f000 0x19000    0x19000    com.apple.driver.usb.AppleUSBUserHCI (1) 6FE7E24C-5793-3AFF-AAFF-9DB4F82066BA <22 20
166    0 0xffffff7fa0bbe000 0x3000     0x3000     com.apple.iokit.IOUserEthernet (1.0.1) D95650D2-397D-3A31-A8E5-5B74994EAAC6 <46 7 6
167    0 0xffffff7f965ab000 0x416000   0x416000   com.apple.kext.AMDRadeonX6100HWLibs (1.0) 34A6BF6A-F7B9-3013-8B7D-F0347B90D1FA <14 6
168    0 0xffffff7fa0bd8000 0x3000     0x3000     com.apple.driver.LuaHardwareAccess (1.0.16) DDAC3137-602E-3520-A3E1-C1ACAF357044 <8
169    0 0xffffff7fa144b000 0x8000     0x8000     com.apple.driver.pmtelemetry (1) 1B9A3B60-046F-3858-9A8E-EFFD697C01C0 <8 7 6 5 3 1>
170    0 0xffffff7f9f1aa000 0x2000     0x2000     com.apple.AGDCPluginDisplayMetrics (6.2.2) 60FC04B2-C96B-37B3-8046-31A09912C52C <120
171    0 0xffffff7f9f1bf000 0x9000     0x9000     com.apple.driver.AppleGraphicsDevicePolicy (6.2.2) D05F43EC-C765-339C-97F3-2E1633C9E
172    0 0xffffff7f81b55000 0x3000     0x3000     com.apple.kext.AMDRadeonServiceManager (4.0.2) BDD0535C-3136-3EC8-B391-94DE579D92B8
173    0 0xffffff7f93403000 0x138000   0x138000   com.apple.kext.AMDRadeonX6000 (4.0.2) 90AA06F1-9663-3AA5-A70B-8F8C5BBA41BB <125 115
174    0 0xffffff7f9f110000 0x17000    0x17000    com.apple.fileutil (20.036.15) 4A53F82C-2C04-3148-A67B-9DFC24C5312C <6 5 3 2 1>
175    1 0xffffff7fa14d4000 0x3000     0x3000     com.apple.kext.triggers (1.0) D4AC084E-97BE-37BC-BEB5-BE9A4BD1705D <8 7 6 5 3 1>
176    0 0xffffff7fa13ab000 0x8000     0x8000     com.apple.filesystems.autofs (3.0) 6BADA3B0-3B9B-3385-9465-3902DD9AB114 <175 8 7 6 5
177    1 0xffffff7fa0aef000 0x6000     0x6000     com.apple.driver.X86PlatformShim (1.0.0) E2E5D882-7FAF-3E79-ABD9-CEE57429403D <146 1
178    0 0xffffff7fa0680000 0x2000     0x2000     com.apple.driver.ApplePlatformEnabler (2.7.0d0) A21C2B68-5CCE-3036-80CC-B232460CA1E1
179    0 0xffffff7f9f20f000 0x1d000    0x1d000    com.apple.driver.AGPM (119) 5728C057-EAAC-341C-87D9-81FC917CDCF9 <123 120 117 115 14
180    0 0xffffff8001690000 0x1000     0x1000     com.apple.driver.AppleHIDALSService (1) AA1D018B-A1BB-3EAF-A817-324D308E6BBB <22 5 3
181    2 0xffffff80016e4000 0xd000     0xd000     com.apple.driver.AppleInputDeviceSupport (4400.35) E537E671-F2B3-3A8D-A70F-17BE6143F
182    2 0xffffff8001825000 0x14000    0x14000    com.apple.driver.AppleMultitouchDriver (4400.28) D6F49AEB-08D3-3701-B74F-B243243A253
183    0 0xffffff8001332000 0x7000     0x7000     com.apple.driver.AppleActuatorDriver (4400.28) BEB8D1A8-32D6-313D-B13A-17594045D021
184    1 0xffffff8002863000 0xf000     0xf000     com.apple.driver.IOBluetoothHIDDriver (8.0.3d9) 5AE6FD64-33E9-33E9-8001-B2B1289535B2
185    1 0xffffff80019c8000 0x6000     0x6000     com.apple.driver.AppleHSBluetoothDriver (4030.5) C816C61B-E933-354F-B844-521080A4FD5
186    0 0xffffff80019d1000 0x7000     0x7000     com.apple.driver.AppleTopCaseHIDEventDriver (4030.5) 76F0CBD7-F1A6-3FE5-9BC9-F82A111
187    0 0xffffff8001692000 0x4000     0x4000     com.apple.driver.AppleHIDKeyboard (223) DB52914E-103E-3DA3-B378-03F6CED02A89 <52 7 6
188    0 0xffffff7fa0785000 0x3000     0x3000     com.apple.driver.AudioAUUC (1.70) 2DB3C567-43FA-309A-8388-B6756C05D67F <133 115 14 1
189    0 0xffffff7fa0711000 0x4b000    0x4b000    com.apple.driver.AppleUSBAudio (401.4) B0746A74-3178-33E2-AB73-DD78A4F3748E <133 106
```

```
File system - Apple_APFS (uses the GPT partition scheme)
```

```
DMI - Intel i7 chips support Northbridge DMI's.
```

```
Geekbench scores (since I'm not using linux, Hardinfo is unavailable for mac):
CPU (i7-9750H):
Single-core : 1012
Multi-core: 5304
GPU (AMD Radeon Pro 5300): 36910
```

Task 1

Goal is to write cache optimised code for matrix multiplication. The basic concept is that the continuous addresses would be loaded into the cache. If we don't access them in a sequential manner, we will have cache misses and the cache will load the addresses that are continuous from the newly accessed memory.

Approach-1

```
for (c = 0; c < m; c++)
  {
    for (d = 0; d < q; d++)
    {
      for (k = 0; k < p; k++)
        sum = sum + A[c][k] * B[k][d];

      N[c][d] = sum;
      sum = 0;
    }
  }
```

Here we don't make the best use of accessing sequential memory. This is because we are accessing a row of A and a column of B in the inner most loop. Arrays in C are stored in a row major format. Thus we are skipping several addresses when we access consecutive elements in different columns.

Doing the math, it is a significant number of cache misses and a very unoptimized way of writing code. Consider a matrix of size 4096*4096 for simpler calculations. In order to fill the row of the resultant matrix, we are going to need to access a row of matrix A (4096 memory reads) and all elements of matrix B (4096*4096 memory reads). That is

16,785,408 memory reads (not all in sequential order) in total to compute a single row. We need to compute 4096 such rows. Keep in mind that our cache size is 32kb which can store 8192 integer values in it.

Approach-2

```
for (jj = 0; jj < n; jj += bsize)
  { // for each bsize block
    for (i = 0; i < m; i++)
      for (j = jj; j < min(jj + bsize, q); j++)
        M[i][j] = 0;
    for (kk = 0; kk < n; kk += bsize)
    {
      for (i = 0; i < m; i++)
      { // for each row of A
        // for each column of the block of B
        for (j = jj; j < min(jj + bsize, q); j++)
        {
          sum = 0;
          // For each element of the sliver of A/column of B
          for (k = kk; k < min(kk + bsize, n); k++)
            sum += A[i][k] * B[k][j];

          M[i][j] += sum;
        }
      }
    }
  }
```

This code takes into account a block size that is equal to the system cache size. Thus the cache misses would be lesser. But it is observed that the code takes longer to run than approach-1 when we don't optimize it during compilation. Though we have optimized for the cache we see that the computation might take longer. However, when we compile it with the -O3 flag, it takes lesser time to execute. Although not that significant.

Approach-3

```
for (ih = 0; ih < m; ih += s)
{
  for (jh = 0; jh < q; jh += s)
  {
    M[ih][jh] = 0;
    for (kh = 0; kh < n; kh += s)
      for (il = 0; il < s; il++)
        for (kl = 0; kl < s; kl++)
          for (jl = 0; jl < s; jl++)
            M[ih + il][jh + jl] += A[ih + il][kh + kl] * B[kh + kl][jh + jl];
  }
}
```

This code optimizes cache better than the other two and has a significantly lower computation time. The logic is that we break the matrix into pieces and compute them separately. This way, we don't have to access more elements at a time than our cache size. 's' is a parameter here. Upon experimenting with various values, I found 32 to deliver the best results. What 's' signifies is the size of the block of the resultant matrix that is calculated. So the block size would be s*s.

To explain it in simpler words. Consider a matrix of size 4096*4096 for simpler calculations. If we set s=32, we need to access s*s elements in the resultant matrix (1024 memory reads), s rows in matrix A (32*4096 = 131,072 memory reads) and s columns in matrix B (4096*32 = 131,072 memory reads). That is a total of 263,168 memory reads.

The advantage of computing it on blocks is to tune the parameter such that it fits in your cache. Furthermore, we can build on this by introducing parallel processing. Since The computing of a cell of the resultant matrix doesn't depend on any other cells of the same, and only on matrices A, B whose values don't change, parallel processing is viable.

Statistics

The following statistics are run on two matrices A, B of size 3500*3500. Their values lie between -600 and 600. They are compiled using:

```
gcc -O3 <file_name>.c
```

Approach comparison

Aa Approach ID	# Time taken to execute (s)	# L1 Cache Misses
1	208.774441	10.5%
2	208.381094	5%
3	4.633474	0.5%

Task 2

Approach-1

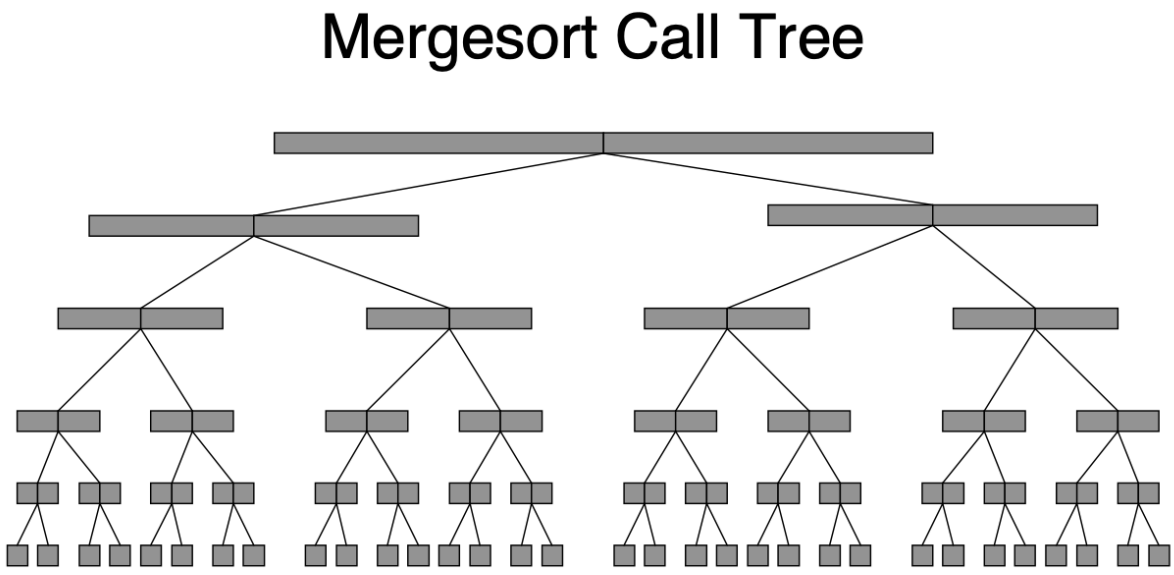
```
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int m = l + (r - l) / 2;

        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        merge(arr, l, m, r);
    }
}
```

We use merge sort as a recursive function. This will divide the array into half and the resultant array into half and so on. Once it reaches the smallest block size, it'll begin to sort and merge them back into the original array. Since the last function call will be executed first, It'll begin by sorting every two consecutive elements and then every four consecutive and so on.

While the access of elements is sequential in a given function call, the number of calls is simply many. Furthermore, it will perform the same operations irrespective of the sorted state of the array provided. That is, it will access roughly the same number of memory states irrespective of how the array is sorted to begin with.

Since the algorithm accesses consecutive elements already, the cache hits are pretty decent. The time taken for a compiler-optimized code (compiled using -O3 flag) gives us good results too.

The working of recursive merge sort in a pictorial form:



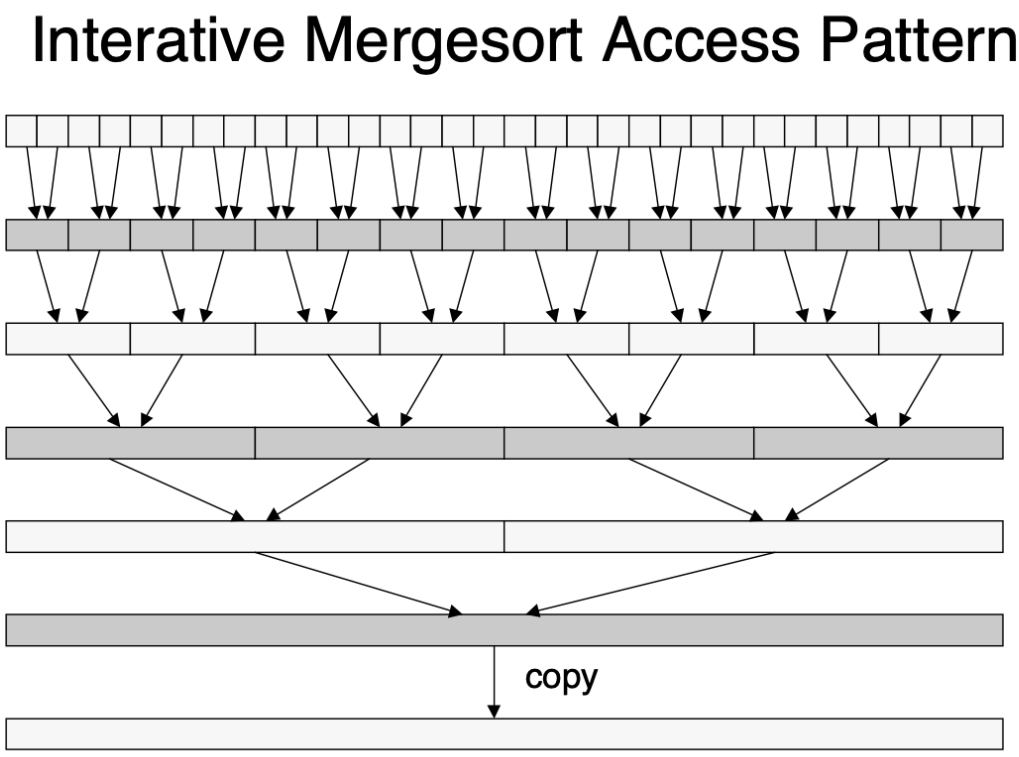
Approach-2

```
for (curr_size = 1; curr_size <= n - 1; curr_size = 2 * curr_size)
{
    for (left_start = 0; left_start < n - 1; left_start += 2 * curr_size)
    {
        int mid = min(left_start + curr_size - 1, n - 1);
        int right_end = min(left_start + 2 * curr_size - 1, n - 1);

        merge(arr, left_start, mid, right_end);
    }
}
```

Instead of approaching merge sort in a recursive manner, I tried it with an iterative method. Thought behind this approach was to eliminate the overhead of several recursive calls from the stack and see if that improves the cache hits.

The results of this are rather surprising. It takes lesser time to execute (by a very small margin) but has higher miss rates. The reason for higher miss rates is that in every iteration of iterative mergesort we access all elements of the array. When the size of the array is large, the data gets unloaded from the cache after manipulations. This means that we need to reload that data into the cache to be able to access it. In contrast, in recursive mergesort, there will always come a time when the function divides the array into pieces smaller than or equal to the size of the cache. My cache can hold up to 8,192 = 2^13 integer elements. Doing the math, recursive mergesort will call the function recursively 2^14 - 1 number of times (16,383). Hence, at max, these many iterations would be cache optimized in recursive mergesort.

The working of iterative merge sort in a pictorial form:



In approaches 1, 2, the drawback is that when the array size crosses the cache size there will be misses. While sorting a very large array, however, this is bound to happen. The most cache-optimal solution would be to use k-way merge sort. The above two are implementations of a 2-way merge sort, where k = cache size (when we say cache size, it is the value of cache size divided by the size of the data type of the element). However, implementing such logic is very difficult. We will look at a 3-way merge sort.

Approach-3

```
int RUN = cache_size / sizeof(arr[0]);

for (i = 0; i < n; i += RUN)
    insertionSort(arr, i, min((i + RUN - 1), (n - 1)));

for (size = RUN; size < n; size = 2 * size)
{
    for (left = 0; left < n; left += 2 * size)
    {
        int mid = left + size - 1;
        int right = min((left + 2 * size - 1),(n - 1));

        if (mid < right)
            merge(arr, left, mid, right);
    }
}
```

This algorithm is widely known as Tim Sort. It segments the entire array into pieces of size of RUN (RUN is a parameter - can be tweaked). It sorts these pieces individually and then merges them. Insertion sort accesses the elements sequentially. Moreover, if the array is already sorted, it need not perform any operations on the array. We've seen one of the major drawbacks of merge sort to be that it performs approximately the same number of operations irrespective of the sorted state of the array. Thus the worst case, best case and average case for merge sort are all equal. While in case of insertion sort, the worst case in O(n^2) while best case is O(n). Thus the average time complexity is O(n).

We already know that in order to achieve the most cache-hits we should access the number of consequent elements equal to our cache size. Thus that is how we set our parameter - RUN. Note: cache size of my machine is 32KB → 8192 integer elements.

Parameter tweaking

Aa Parameter Value (RUN)	# Time taken to execute (s)	# D1 Cache Misses	# LLd Cache Misses
16	0.499073	3.4%	1.8%
32	0.521262	3.2%	1.6%
64	0.487582	2.8%	1.4%
128	0.51153	2.2%	1.1%
512	0.712436	0.9%	0.4%
8192	4.159793	0.1%	0%

In reality, it turns out the most cache-optimized code is the slowest. This is because insertion sort works best on small-sized arrays. As the size of the array increases, the worst-case time complexity dominates. The best performance is observed when the parameter is set to 64.

Approach-4

```
void mergeSort3WayRec(int gArray[], int low, int high, int destArray[])
{
    if (high - low < 2)
        return;

    int mid1 = low + ((high - low) / 3);
    int mid2 = low + 2 * ((high - low) / 3) + 1;

    mergeSort3WayRec(destArray, low, mid1, gArray);
    mergeSort3WayRec(destArray, mid1, mid2, gArray);
    mergeSort3WayRec(destArray, mid2, high, gArray);

    merge3(destArray, low, mid1, mid2, high, gArray);
}
```

The reason we are looking at this approach at the end is to understand how the cache optimization is done here. We mentioned how the maximum number of cache optimized iterations 2-way mergesort can do. Now let us take a look at how 3-way mergesort works. It will divide the array into three pieces, sort them, and merge them. The advantage over 2-way mergesort is that we will always have a small array to deal with. This helps it fit in our cache better. However,

there is a complication. That being, we cannot go beyond an array of size 2e6 since the number of stack calls it would require would cause a segmentation error. We have already seen that the overhead of calling recursive functions from the stack is negligible when we have the power of loading our input directly from our cache. Thus we observe this approach to be the most cache-friendly and optimized among the others. It is also the fastest. The reason it is faster than Tim sort is simply that insertion sort has a worst-case time complexity of O(n^2) while 3-way merge sort has a complexity of O(log_3(N)).

Statistics

The array is of size 6,000,000 and each element is an integer. The code is compiled using:

```
gcc -O3 <file_name>.c
```

Approach comparison

Approach ID	Time taken to execute (s)	D1 Cache Misses	LLd Cache Misses
1	0.634567	2.4%	0.6%
2	0.624017	3%	1.6%
3	0.487582	2.8%	1.4%

For an array of size 2,000,000 and each element is an integer. The code is compiled the same way as shown above.

Approach comparison

Approach ID	Time taken to execute (s)	D1 Cache Misses	LLd Cache Misses
1	0.196922	0.6%	0.1%
2	0.179542	0.9%	0.1%
3	0.152587	0.8%	0.1%
4	0.05465	0.4%	0.1%

Additional Points

- My system information is provided as TASK-0 and not the server I was allocated on abacus since I did most of the computation on my local machine. I used the server to run valgrind. The cache size of the server and my machine are equal (32KB). Hence, my results would be same if run on my machine.
- In my report, the time taken to execute is the time taken by the code post compiler optimization. This was done since it took too long to execute normally and that meant more waiting and less testing.
- Input is taken via file. Followed the convention given in the assignment PDF. The naming convention followed is 2_input.txt and 3_input.txt.
- Output is via terminal and also saved in a file. The files generated will be 2_output.txt and 3_output.txt.
- In the matrix multiplication, there is a discrepancy between the variables described in the PDF and the code. I have corrected it according to the PDF.
- In addition to my report and the two codes, I'm also attaching some test cases to show how I've used the input and output.