Linux For Embedded Systems

For Arabs

Course 102:

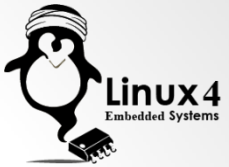
Understanding Linux

Ahmed ElArabawy

Linux4
Embedded Systems

Lecture 28:
Virtual FileSystems

A Virtual FileSystem



- A virtual filesystem is a filesystem that resides in memory and does not have physical files stored on some storage device
- Instead, it is just a way to send information between the user application and the kernel in the form of reading a file or writing to a file
- Examples of virtual filesystems are
  - The procfs filesystem which is mounted in **/proc**
  - The sysfs filesystem which is ususally mounted in **/sys**
- When we write to a virtual filesystem file (as in **/proc** or **/sys**), we are sending information to the kernel for processing, and not storing data in a file
- When we are reading from a file in a virtual filesystem, we are actually querying the kernel, and causing some processing with some output, and not just reading from a file
- That is why, when we list files in a virtual filesystem, we normally find the timestamp showing the current time, and the file size maybe zero although we get data when we perform a read (through cat command for example)

A Window to the Kernel

- The **procfs** and **sysfs** filesystems are one way to learn a lot of information about the kernel by user space applications
- Most files in **/proc** are read only but some are writable (specially in **/proc/sys/**)
- There are more files that are writable in **/sys**

A Treasure of Information

- There are a lot of information that you can read from the **/proc** and **/sys**
- Information about processes, memory, interrupts, filesystems, hardware devices, network, and much more
- The initial role for **/proc** was to carry information about the processes, but it ended up with much more information
- The initial role for **/sys** was to carry information about the system devices and hardware buses but it also ended up with much more info

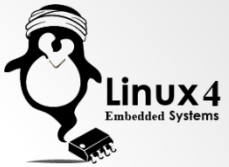
The Unknown Soldier

- A lot of commands that we worked with in previous lectures does nothing except reading information from **/proc** and **/sys** and print it after some organization
- Examples are : **ps**, **top**, **mount**, **uptime**, …
- If you unmount **/proc** or **/sys**, a lot of commands will fail to perform its job

Linux4
Embedded Systems

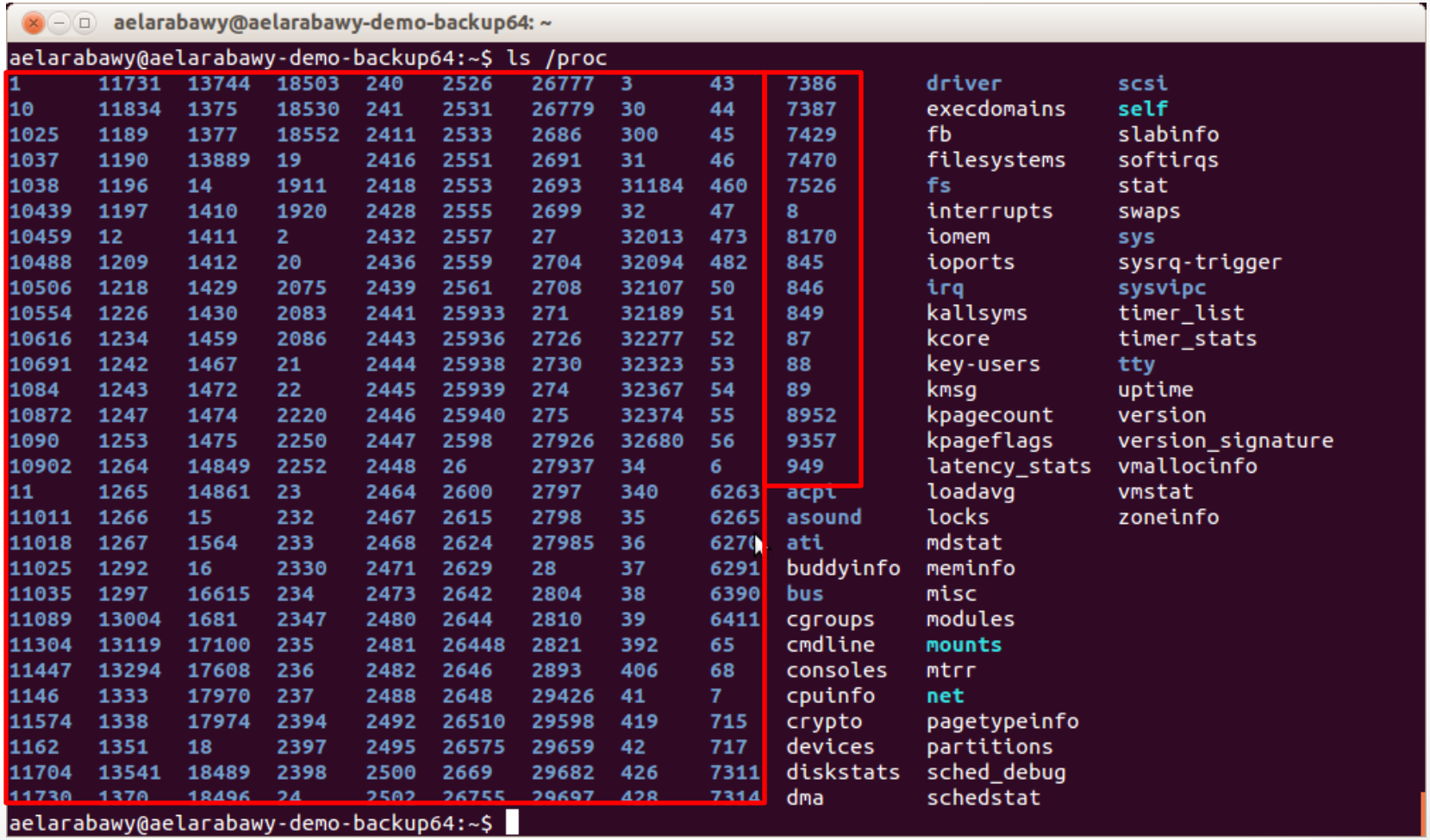
/proc

Listing /proc



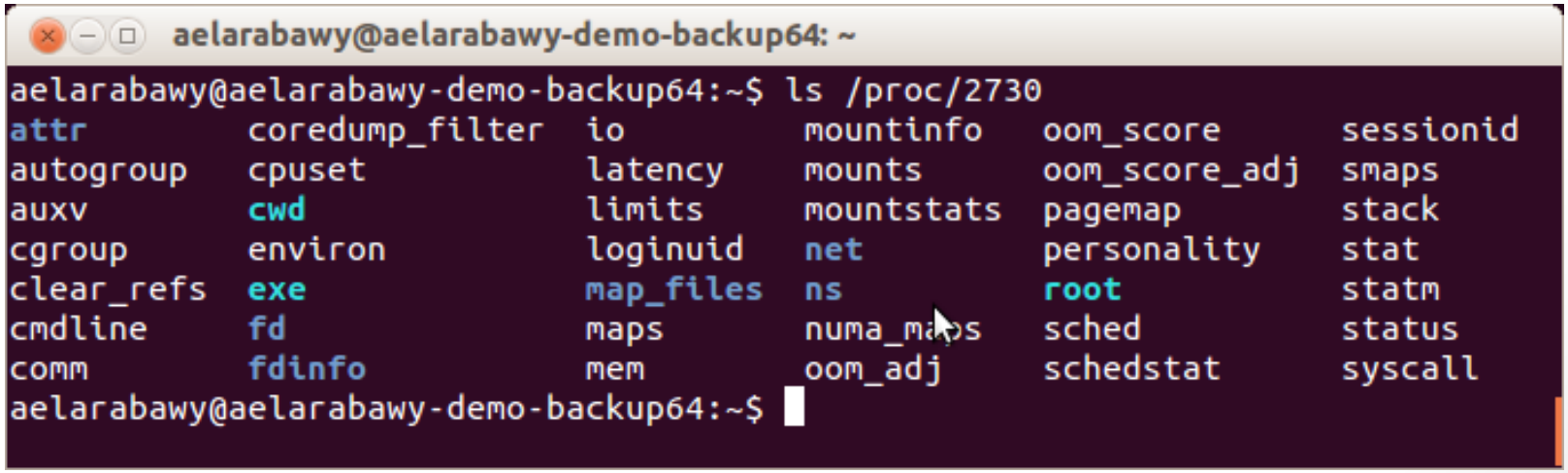

```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ls /proc
1       11731  13744  18503  240   2526   26777  3      43    7386       driver          scsi
10      11834  1375   18530  241   2531   26779  30     44    7387       execdomains     self
1025    1189   1377   18552  2411  2533   2686   300    45    7429       fb              slabinfo
1037    1190   13889  19     2416  2551   2691   31     46    7470       filesystems     softirqs
1038    1196   14     1911   2418  2553   2693   31184  460   7526       fs              stat
10439   1197   1410   1920   2428  2555   2699   32     47    8          interrupts      swaps
10459   12     1411   2      2432  2557   27     32013  473   8170       iomem           sys
10488   1209   1412   20     2436  2559   2704   32094  482   845        ioports         sysrq-trigger
10506   1218   1429   2075   2439  2561   2708   32107  50    846        irq             sysvipc
10554   1226   1430   2083   2441  25933  271    32189  51    849        kallsyms        timer_list
10616   1234   1459   2086   2443  25936  2726   32277  52    87         kcore           timer_stats
10691   1242   1467   21     2444  25938  2730   32323  53    88         key-users       tty
1084    1243   1472   22     2445  25939  274    32367  54    89         kmsg            uptime
10872   1247   1474   2220   2446  25940  275    32374  55    8952       kpagecount      version
1090    1253   1475   2250   2447  2598   27926  32680  56    9357       kpageflags      version_signature
10902   1264   14849  2252   2448  26     27937  34     6     949        latency_stats   vmallocinfo
11      1265   14861  23     2464  2600   2797   340    6263  acpi       loadavg         vmstat
11011   1266   15     232    2467  2615   2798   35     6265  asound     locks           zoneinfo
11018   1267   1564   233    2468  2624   27985  36     627(  ati        mdstat
11025   1292   16     2330   2471  2629   28     37     6291  buddyinfo  meminfo
11035   1297   16615  234    2473  2642   2804   38     6390  bus        misc
11089   13004  1681   2347   2480  2644   2810   39     6411  cgroups    modules
11304   13119  17100  235    2481  26448  2821   392    65    cmdline    mounts
11447   13294  17608  236    2482  2646   2893   406    68    consoles   mtrr
1146    1333   17970  237    2488  2648   29426  41     7     cpuinfo    net
11574   1338   17974  2394   2492  26510  29598  419    715   crypto     pagetypeinfo
1162    1351   18     2397   2495  26575  29659  42     717   devices    partitions
11704   13541  18489  2398   2500  2669   29682  426    7311  diskstats  sched_debug
11730   1370   18496  24     2502  26755  29697  428    7314  dma        schedstat
aelarabawy@aelarabawy-demo-backup64:~$
```

Process List

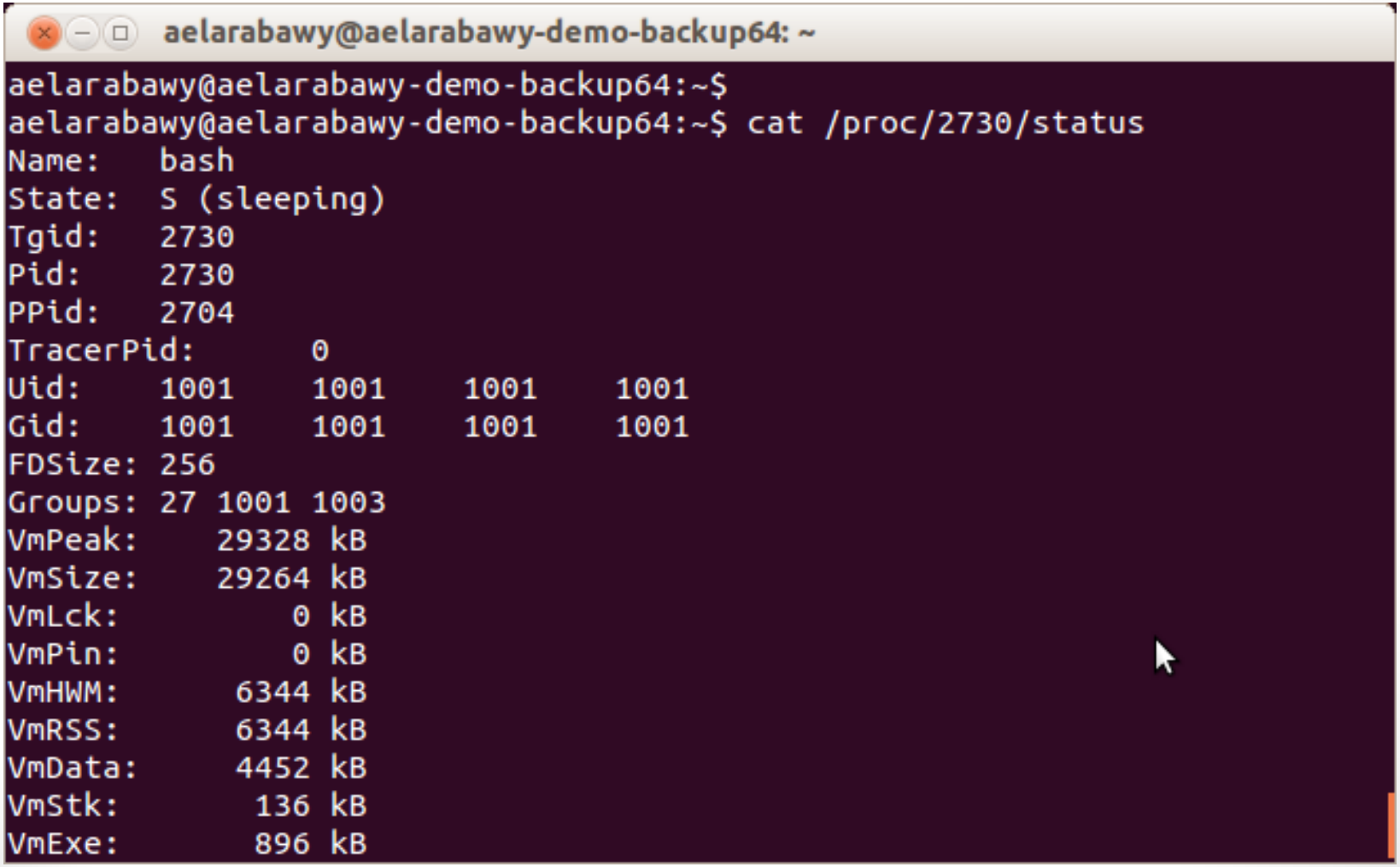


Process List

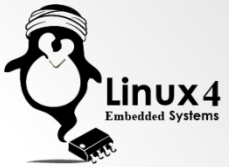
- The /proc contains a directory for each process running on the system
- Directories for processes are named with the process pid
- For example the **init** process will be represented by **/proc/1**
- Now looking inside each process directory



/proc/<pid>/status

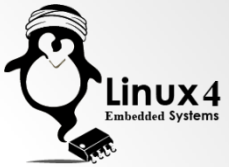


/proc/cpuinfo



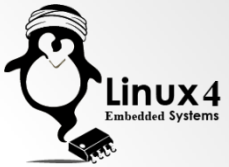
- This file will carry the information about the processor cores in the machine
- It will show a list of all the processor cores with their attributes
  - Core number
  - Processor name, family, and model name
  - Cache Size
  - Does it have a FPU
  - Processor Speed in MHz

/proc/interrupts



- This file will report information on the processor interrupts
- It will have a table for interrupts for all the cores in the system to show the number of interrupts for every interrupt line on each processor, and how many times this interrupt happened
- It will also show which devices are servicing these interrupts

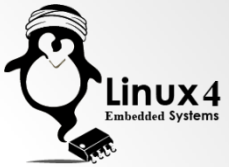
/proc/modules



- This file will list all the KLM (Kernel Loadable Modules) that are loaded in the system
- The contents of this file are used by the command

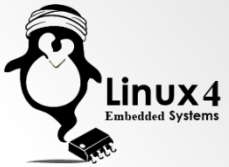
  $ lsmod

/proc/cmdline



- This file shows the command line arguments that where passed to the kernel at its startup
- It will state things like
  - Location of the root file system
  - Where does the kernel send its messages (the different consoles)
  - Should the kernel be “quiet” which means, it will not send its output to a console
  - Location of the boot image for the kernel

/proc/kcore

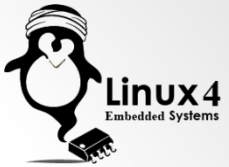


- This file represents all the physical memory of the system
- Size of this file is same as the physical memory + 4 bytes
- Don’t try to list the file content since it is huge and will probably hang your system
- It can be used with debugger to analyze memory issues during development

Linux4
Embedded Systems

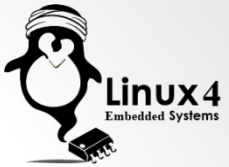
/sys

/sys



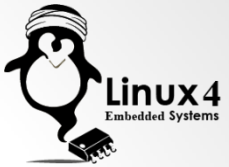
- The **sysfs** is similar to the **procfs** from the perspective that it is a virtual filesystem, and reading/writing to its files trigger functionality in the kernel
- However, it comes with some differences,
  - It mainly targets the description of attributes of system devices and hardware (both configuration and statistics)
  - Its output/input is formatted to be used by a program, and hence it is not very human readable in some cases
  - Each file represents one value (whether input or output)
    - It does not have files that carry tables of information, or a list of parameters.
    - If you need to show 4 counter statistics, then you will have 4 readable files in **/sys** (each file will contain a single counter value)
    - If you need to configure two attributes of a device, you will need 2 writable files in /sys

Listing /sys




```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$
aelarabawy@aelarabawy-demo-backup64:~$ ls /sys
block  bus  class  dev  devices  firmware  fs  hypervisor  kernel  module  power
aelarabawy@aelarabawy-demo-backup64:~$ ls /sys/devices
breakpoint  LNXSYSTM:00  platform  rapidio   system      virtual
cpu         pci0000:00   pnp0      software  tracepoint
aelarabawy@aelarabawy-demo-backup64:~$ ls /sys/class
ata_device  bsg       hidraw       misc          printer       scsi_host      tty
ata_link    devfreq   hwmon        mmc_host      regulator     sound          vc
ata_port    dma       i2c-adapter  net           rfkill        spi_host       vtconsole
backlight   dmi       input        pci_bus       rtc           spi_master     watchdog
bdi         firmware  leds         power_supply  scsi_device   spi_transport
block       gpio      mdio_bus     ppdev         scsi_disk     thermal
bluetooth   graphics  mem          ppp           scsi_generic  timed_output
aelarabawy@aelarabawy-demo-backup64:~$ ls /sys/block
loop0  loop2  loop4  loop6  ram0  ram10  ram12  ram14  ram2  ram4  ram6  ram8  sda
loop1  loop3  loop5  loop7  ram1  ram11  ram13  ram15  ram3  ram5  ram7  ram9  sr0
aelarabawy@aelarabawy-demo-backup64:~$ ls /sys/kernel
debug    fscaps              kexec_crash_size  mm     profiling  slab           uevent_seqnum
fscache  kexec_crash_loaded  kexec_loaded      notes  security   uevent_helper  vmcoreinfo
aelarabawy@aelarabawy-demo-backup64:~$
```

One Value Per File




```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$
aelarabawy@aelarabawy-demo-backup64:~$
aelarabawy@aelarabawy-demo-backup64:~$ ls /sys/class/net/eth1/statistics/
collisions     rx_dropped        rx_missed_errors   tx_carrier_errors  tx_heartbeat_errors
multicast      rx_errors         rx_over_errors     tx_compressed      tx_packets
rx_bytes       rx_fifo_errors    rx_packets         tx_dropped         tx_window_errors
rx_compressed  rx_frame_errors   tx_aborted_errors  tx_errors
rx_crc_errors  rx_length_errors  tx_bytes           tx_fifo_errors
aelarabawy@aelarabawy-demo-backup64:~$ cat /sys/class/net/eth1/statistics/rx_bytes
27000255861
aelarabawy@aelarabawy-demo-backup64:~$ cat /sys/class/net/eth1/statistics/tx_bytes
4551221053
aelarabawy@aelarabawy-demo-backup64:~$ cat /sys/class/net/eth1/statistics/tx_errors
0
aelarabawy@aelarabawy-demo-backup64:~$ cat /sys/class/net/eth1/statistics/tx_packets
7127126
aelarabawy@aelarabawy-demo-backup64:~$ cat /sys/class/net/eth1/statistics/rx_packets
24240585
aelarabawy@aelarabawy-demo-backup64:~$ cat /sys/class/net/eth1/statistics/rx_dropped
0
aelarabawy@aelarabawy-demo-backup64:~$
```

Linux 4 Embedded Systems

http://Linux4EmbeddedSystems.com