The Android Linux Kernel (Part 1):
Primary & Secondary Storage Mechanisms

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

1. Recognize the two types of storage supported by Android Linux: *primary & secondary storage*
Overview of Android Linux
Primary & Secondary Storage Mechanisms
Android Linux Kernel: Primary & Secondary Storage

- Android's software instructions & data reside in two different types of storage

See [en.wikipedia.org/wiki/Computer_data_storage](en.wikipedia.org/wiki/Computer_data_storage)
• Android’s software instructions & data reside in two different types of storage

• **Primary storage** – Fast random access memory (RAM)

See [en.wikipedia.org/wiki/Computer_data_storage#Primary_storage](en.wikipedia.org/wiki/Computer_data_storage#Primary_storage)
Android’s software instructions & data reside in two different types of storage

- **Primary storage** – Fast random access memory (RAM)
  - The contents of volatile RAM are wiped out whenever a device is rebooted or loses power

See en.wikipedia.org/wiki/Volatile_memory
Android Linux Kernel: Primary & Secondary Storage

- Android’s software instructions & data reside in two different types of storage
  - **Primary storage** – Fast random access memory (RAM)
  - **Secondary storage** – Slower flash memory

Android’s software instructions & data reside in two different types of storage

- **Primary storage** – Fast random access memory (RAM)
- **Secondary storage** – Slower flash memory

Flash is persistent storage that can be erased & reprogrammed electronically

See [en.wikipedia.org/wiki/Flash_memory](en.wikipedia.org/wiki/Flash_memory)
Primary & secondary storage are part of a “memory hierarchy”

See en.wikipedia.org/wiki/Memory_hierarchy
• In general, being “higher” in this hierarchy has several implications

See en.wikipedia.org/wiki/Memory_bandwidth
In general, being “higher” in this hierarchy has several implications, e.g.
- Memory bandwidth is faster
  - i.e., rate at which data can be read from or stored into

See en.wikipedia.org/wiki/Memory_bandwidth
In general, being “higher” in this hierarchy has several implications, e.g.:

- Memory bandwidth is faster
- CPU access latency is lower
  - i.e., time interval between when CPU initiates a call for fetch or store data & when the call completes

See en.wikipedia.org/wiki/Access_time
• In general, being “higher” in this hierarchy has several implications, e.g.
  • Memory bandwidth is faster
  • CPU access latency is lower
  • Cost is greater
    • i.e., “faster” == “costlier”

Android Linux Kernel: Primary & Secondary Storage

- Processor cores operate on instructions & their associated data that reside in RAM

See en.wikipedia.org/wiki/Central_processing_unit#Operation
• Processor cores operate on instructions & their associated data that reside in RAM
• Android Linux executes in kernel space RAM

See www.linfo.org/kernel_space.html
- Processor cores operate on instructions & their associated data that reside in RAM
- Android Linux executes in kernel space RAM
  - Kernel space is a protected region of RAM for running privileged operations

See en.wikipedia.org/wiki/CPU_modes
• Processor cores operate on instructions & their associated data that reside in RAM

• Android Linux executes in kernel space RAM
  • Kernel space is a protected region of RAM for running privileged operations
  • Kernel space can be accessed by user processes only via system calls

See en.wikipedia.org/wiki/System_call
• Processor cores operate on instructions & their associated data that reside in RAM
  • Android Linux executes in kernel space RAM
  • All Android apps execute in user space RAM

See en.wikipedia.org/wiki/User_space
• Processor cores operate on instructions & their associated data that reside in RAM
  • Android Linux executes in kernel space RAM
  • All Android apps execute in user space RAM
    • User space is a more restrictive protection domain than kernel space

See www.linfo.org/user_space.html
• Processor cores operate on instructions & their associated data that reside in RAM
  • Android Linux executes in kernel space RAM
  • All Android apps execute in user space RAM
    • User space is a more restrictive protection domain than kernel space
    • Apps running in user space normally can’t access RAM of other apps, unless explicitly allowed

See “anonymous shared memory” discussion in part 3 of this lesson
• The cost & performance of primary & secondary storage has improved substantially in recent years

See en.wikipedia.org/wiki/Random-access_memory#Recent_developments
Primary storage (RAM) on Android mobile devices is constrained.

See developer.android.com/training/articles/memory.html
Primary storage (RAM) on Android mobile devices is constrained, e.g.

- Form factor

In 2017 2-4 GB is common for mobile devices versus 8-64 GB on a desktop or laptop.
• Primary storage (RAM) on Android mobile devices is constrained, e.g.
  • Form factor
  • Price points

Lower cost mobile devices typically have much less RAM than higher cost devices
• (Android) Linux’s virtual memory manager features address memory constraints

See [en.wikipedia.org/wiki/Virtual_memory](en.wikipedia.org/wiki/Virtual_memory)
(Android) Linux’s virtual memory manager features address memory constraints, e.g.

- Helps conserve RAM by not moving app instructions & data from secondary to primary storage until they are accessed.
(Android) Linux’s virtual memory manager features address memory constraints, e.g.

- Helps conserve RAM by not moving app instructions & data from secondary to primary storage until they are accessed
- Accelerates I/O operations via memory mapping of files & hardware devices

See [en.wikipedia.org/wiki/Memory-mapped_file](en.wikipedia.org/wiki/Memory-mapped_file)
(Android) Linux’s virtual memory manager features address memory constraints, e.g.

- Helps conserve RAM by not moving app instructions & data from secondary to primary storage until they are accessed
- Accelerates I/O operations via memory mapping of files & hardware devices
- Protects an app’s private data in RAM from other apps

See en.wikipedia.org/wiki/Memory_management_unit
Secondary storage in Android Linux is used to save data persistently.

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>Persistent Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Preferences</td>
<td>Store private primitive data in key-value pairs</td>
</tr>
<tr>
<td>External Storage</td>
<td>Store public data on the shared external storage</td>
</tr>
<tr>
<td>Internal Storage</td>
<td>Store private data on the device memory</td>
</tr>
<tr>
<td>SQLite Databases</td>
<td>Store structured data in a private database</td>
</tr>
</tbody>
</table>

See developer.android.com/guide/topics/data/data-storage.html
Secondary storage in Android Linux is used to save data persistently.

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>Persistent Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Preferences</td>
<td>Store private primitive data in key-value pairs</td>
</tr>
<tr>
<td>External Storage</td>
<td>Store public data on the shared external storage</td>
</tr>
<tr>
<td>Internal Storage</td>
<td>Store private data on the device memory</td>
</tr>
<tr>
<td>SQLite Databases</td>
<td>Store structured data in a private database</td>
</tr>
</tbody>
</table>
Secondary storage in Android Linux is used to save data persistently.

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>Persistent Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Preferences</td>
<td>Store private primitive data in key-value pairs</td>
</tr>
<tr>
<td>External Storage</td>
<td>Store public data on the shared external storage</td>
</tr>
<tr>
<td>Internal Storage</td>
<td>Store private data on the device memory</td>
</tr>
<tr>
<td>SQLite Databases</td>
<td>Store structured data in a private database</td>
</tr>
</tbody>
</table>

Android has been progressively enhancing security of external storage over time.
Secondary storage in Android Linux is used to save data persistently

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>Persistent Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Preferences</td>
<td>Store private primitive data in key-value pairs</td>
</tr>
<tr>
<td>External Storage</td>
<td>Store public data on the shared external storage</td>
</tr>
<tr>
<td>Internal Storage</td>
<td>Store private data on the device memory</td>
</tr>
<tr>
<td>SQLite Databases</td>
<td>Store structured data in a private database</td>
</tr>
</tbody>
</table>
Secondary storage in Android Linux is used to save data persistently.

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>Persistent Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Preferences</td>
<td>Store private primitive data in key-value pairs</td>
</tr>
<tr>
<td>External Storage</td>
<td>Store public data on the shared external storage</td>
</tr>
<tr>
<td>Internal Storage</td>
<td>Store private data on the device memory</td>
</tr>
<tr>
<td>SQLite Databases</td>
<td>Store structured data in a private database</td>
</tr>
</tbody>
</table>
Android Linux supports secondary storage via its Virtual File System (VFS) framework.

See www.all-things-android.com/content/understanding-android-file-hierarchy
Android Linux supports secondary storage via its Virtual File System (VFS) framework. Each file system is implemented via a kernel module that registers the operations that it supports with VFS.

See en.wikipedia.org/wiki/Loadable_kernel_module
Android Linux supports secondary storage via its Virtual File System (VFS) framework.

Each file system is implemented via a kernel module that registers the operations that it supports with VFS.

Android Linux file systems support “flash memory” files that can be erased/reprogrammed electronically.

See [en.wikipedia.org/wiki/Flash_memory](en.wikipedia.org/wiki/Flash_memory)
End of the Android Linux Kernel: (Part 1) Primary & Secondary Storage Mechanisms