Android is NOT just 'Java on Linux'

2011.5.22
2011.10.26 updated

Tetsuyuki Kobayashi

Let's talk about inside of Android.

http://www.kmckk.co.jp/eng/kzma9/
http://www.kmckk.co.jp/eng/jet_index.html

Who am I?

- 20+ years involved in embedded systems
  - 10 years in real time OS, such as iTRON
  - 10 years in embedded Java Virtual Machine
  - Now GCC, Linux, QEMU, Android, …
- Blogs
  - http://d.hatena.ne.jp/embedded/ (Personal)
  - http://blog.kmckk.com/ (Corporate)
  - http://kobablog.wordpress.com/(English)
- Twitter
  - @tetsu_koba

Android is NOT just 'Java on Linux'

- Android uses Linux kernel. Only kernel.
  - User land is totally different from usual Linux system.
- Android applications are written in Java language.
  - Class libraries are similar to Java SE but not equal.
- Dalvik VM eats only dex code
  - need to translate from Java byte code in advance

Let's explore inside of Android

- Assuming you know Linux and Java very well :)

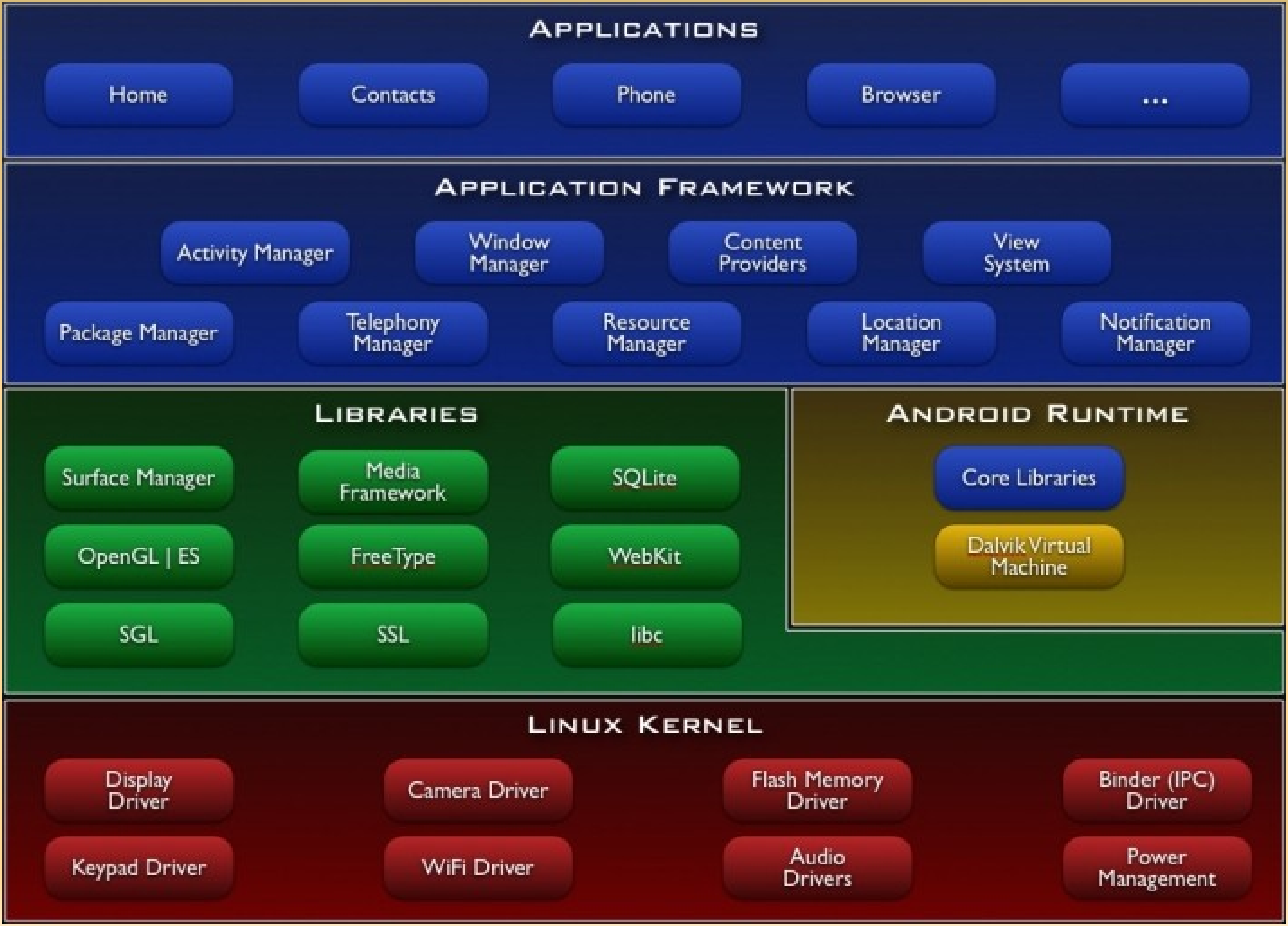
Today's topic

- Android system architecture
- Init – runtime – Zygoto
- Dalvik VM
- Android specific kernel drivers
- How to build Android

Today's topic

- **Android system architecture**
- Init – runtime – Zygoto
- Dalvik VM
- Android specific kernel drivers
- How to build Android

System architecture



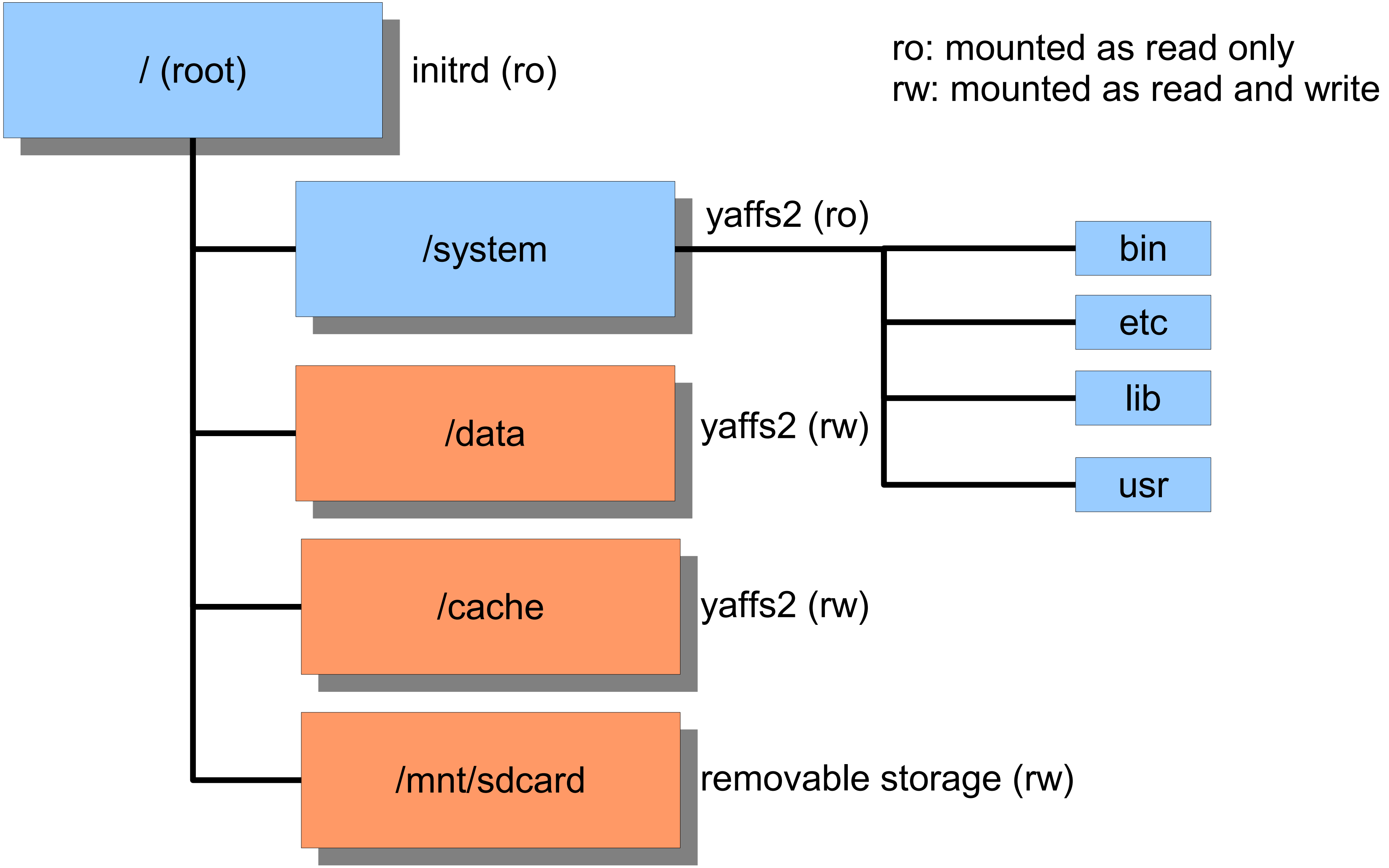
Java is the first class citizen in Android

- Dalvik VM is the center of Android runtime.
- Almost all daemon services are written in Java.
- Application life cycle is described by Java API

Java is the first class citizen in Android

- NDK
  - native library called from Java via JNI
  - This is just a library. Application life cycle is the same as Java.
- Native activity
  - Only C/C++ to make Apps. (just hidden JNI part into system.)
  - not short-cut for C/C++

Typical Directory Tree of Android

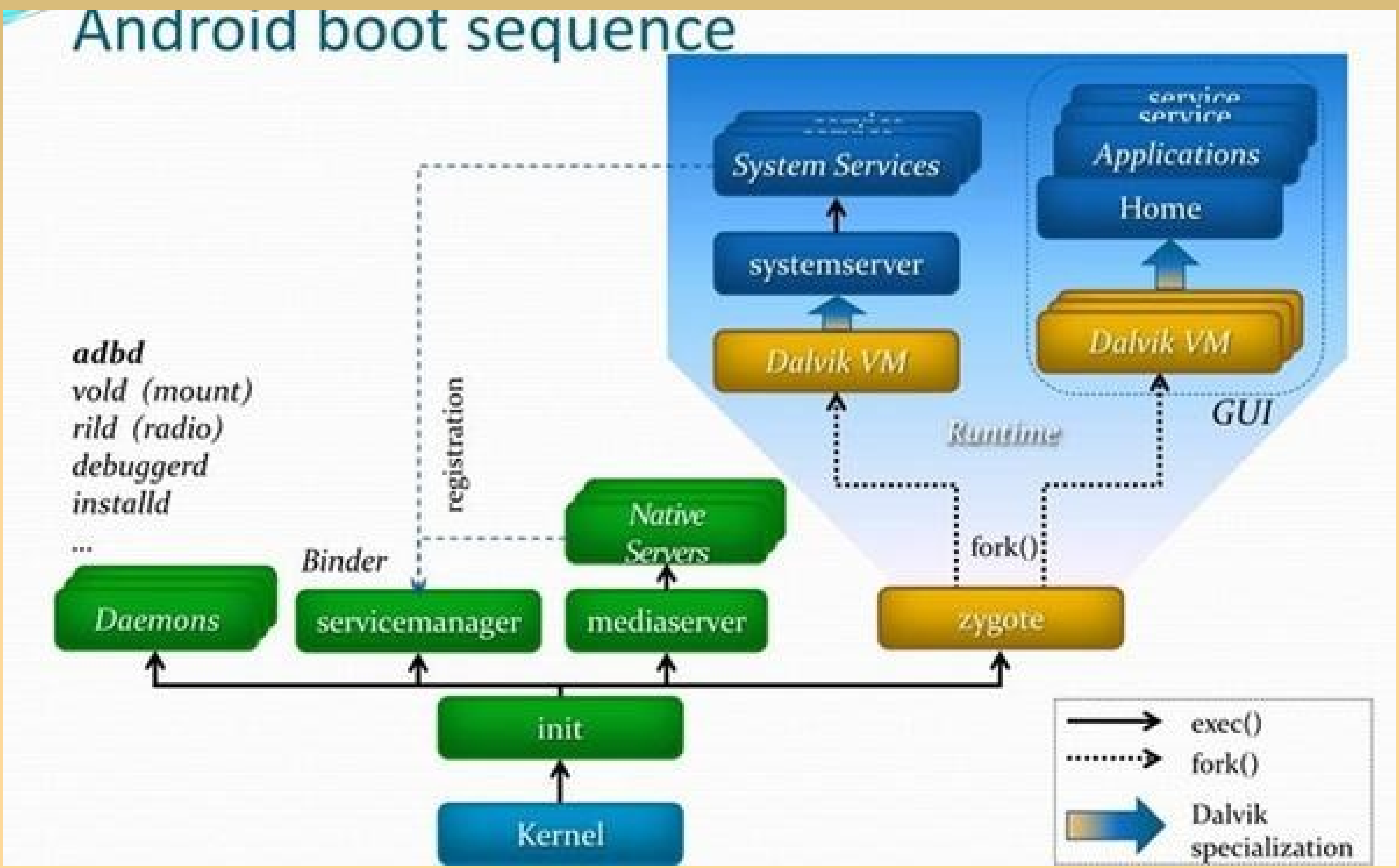


cf. Usual Linux system assumes all file system are read/writable.

Today's topic

- Android system architecture
- **Init – runtime – Zygoto**
- Dalvik VM
- Android specific kernel drivers
- How to build Android

Boot sequence



quoted from http://hmtsay.blogspot.com/2010/10/android-startup.html

init

- located on /init
  - need kernel boot parameter to add “init=/init”
- Static linked.
  - cf. typical linux init is dynamic linked.
  - Doesn't affect even dynamic link system collapsed.
- http://blog.kmckk.com/archives/3137191.html

Bionic

- The standard libraries
  - libc, libm, pthread, dynamic linker
  - linker has implicit crash dump function
    - http://kobablog.wordpress.com/2011/05/12/debuggerd-of-android/
- Came from *BSD, not glibc
- Currently, doesn't support C++ exception and RTTI.
  - latest NDK supports these by static linking.

Prelinking

- Locate dynamic link libraries ahead of time.
- 'apriori' command. Different from 'prelink' command from Red Hat.
- Optimized for small embedded system
  - Allocate fixed address to libraries .
  - Assume not adding/removing libraries.
  - Assume 3GB memory space is large enough to put all libraries together.

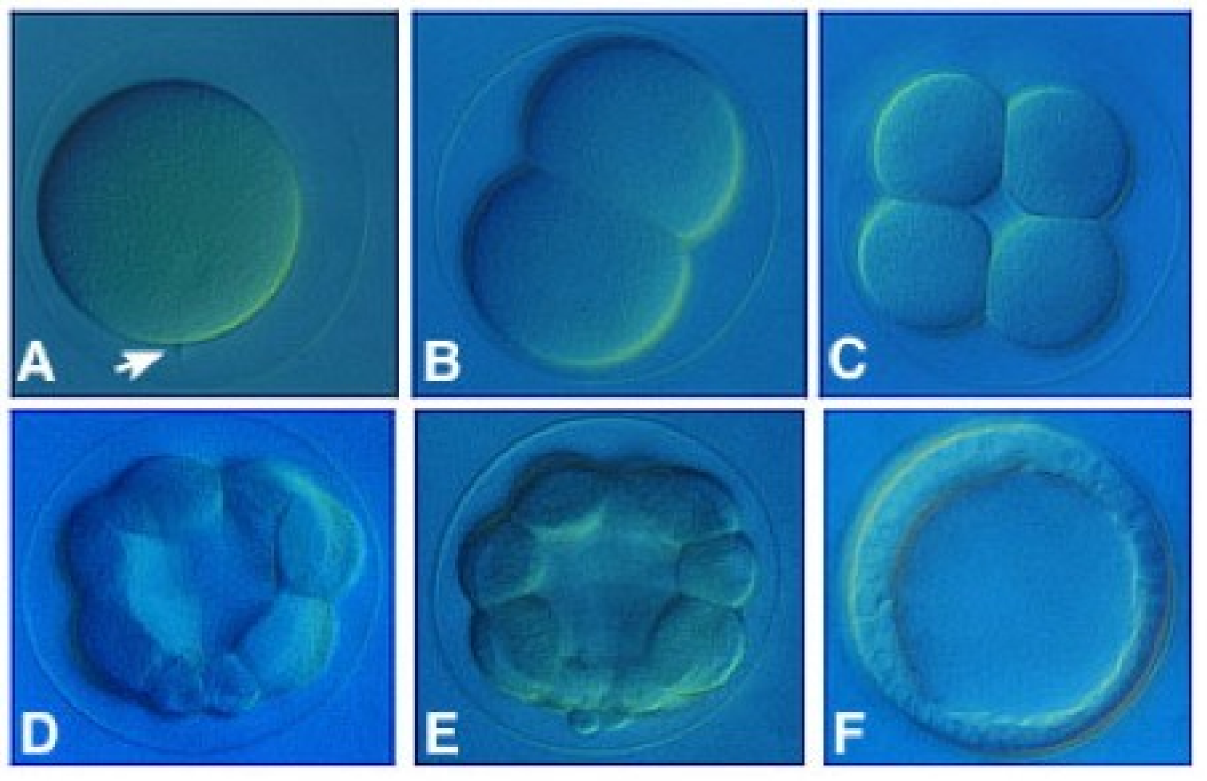
Prelink map

build/core/prelink-linux-arm.map

```
# 0xC0000000 - 0xFFFFFFFF Kernel
# 0xB0100000 - 0xBFFFFFFF Thread 0 Stack
# 0xB0000000 - 0xB00FFFFF Linker
# 0xA0000000 - 0xBFFFFFFF Prelinked System Libraries
# 0x90000000 - 0x9FFFFFFF Prelinked App Libraries
# 0x80000000 - 0x8FFFFFFF Non-prelinked Libraries
# 0x40000000 - 0x7FFFFFFF mmap'd stuff
# 0x10000000 - 0x3FFFFFFF Thread Stacks
# 0x00000000 - 0x0FFFFFFF .text / .data / heap
```

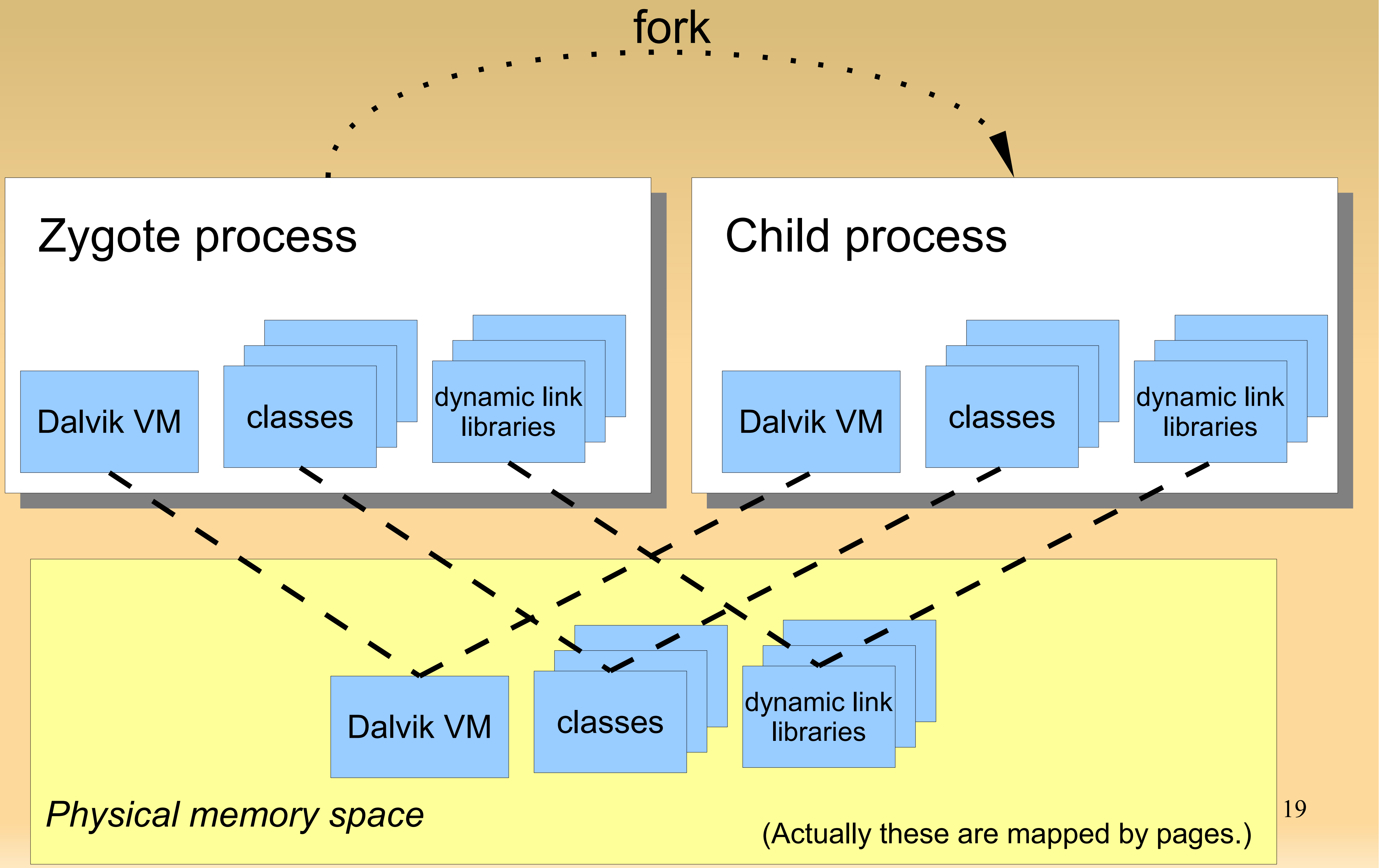
```
# core system libraries
libdl.so                0xAFF00000 # [<64K]
libc.so                 0xAFD00000 # [~2M]
libstdc++.so            0xAFC00000 # [<64K]
libm.so                 0xAFB00000 # [~1M]
liblog.so               0xAFA00000 # [<64K]
libcutils.so            0xAF900000 # [~1M]
libthread_db.so         0xAF800000 # [<64K]
libz.so                 0xAF700000 # [~1M]
libevent.so             0xAF600000 # [???]
libssl.so               0xAF400000 # [~2M]
libcrypto.so            0xAF000000 # [~4M]
libsysutils.so          0xAEF00000 # [~1M]
   ...
```

Zygote



quoted from http://worms.zoology.wisc.edu/dd2/echino/cleavage/intro.html

Zygote



(Actually these are mapped by pages.)

Zygote

- Zygote process preloads typical (approx. 1800) classes and dynamic link libraries so that childlen start quickly.
- Copy-on-write
  - Only when new process writes page, new page is allocated.
  - All pages not be written are shared among all zygote children.
- Exec system call is not used in zygote.
  - Exec wipes the page mapping table of the process.
  - It means exec discards zygote cache.

UID, GID of Applications

- UID(user id) and GID(group id) is used for managing multi-user in usual Linux system.
- Android use this mechanism to isolate applications.
  - Each application has unique UID.
  - Can not read/write other application's files.
- Zygote is running as UID=0 (root). After forking child process, its UID is changed by setuid system call.

Today's topic

- Android system architecture
- Init – runtime – Zygoto
- **Dalvik VM**
- Android specific kernel drivers
- How to build Android

Dalvik VM

- executes dex code, which is translated from Java byte code
- 16bit, register based
  - cf. Java bytecode is 8bit, stack based
- has JIT from Android 2.2 (Froyo)
  - http://blog.kmckk.com/archives/2691473.html
- has concurrent GC from Android 2.3 (Gingerbread)
- http://source.android.com/tech/dalvik/

Java class libraries

- Different from Java ME, which is used in traditional Japanese phone.
- Similar to Java SE. But not equal.
  - Different window/graphics. No AWT, No Swing.
  - No RMI.
- Take care to use user defined class loader
  - dynamic generated classes doesn't work because Dalvik VM doesn't eat Java class files but Dex files.

Caveats of NDK programming

- Dynamic libraries built by NDK are linked with application process.
  - forked from Zygote but UID != 0 (root).
  - consider about permissions.
- Don't use fork & exec system calls.
  - Back ground process should be made as android .app.Service.
- Don't use GCC's TLS extension (__thread).
  - Simple Android dynamic linker does not support it.
  - java.lang.ThreadLocal is available in Java.

3 commands to invoke Dalvik VM

- /system/bin/app_process
  - This is the 'Zygote' process.
- /system/bin/dalvikvm
  - Similar to usual 'java' command.
  - Try 'dalvikvm -h' to show command line help.
- /system/bin/dvz
  - Send request to Zygote process.
- See my blog (Sorry in Japanese)
  - http://blog.kmckk.com/archives/3551546.html

Today's topic

- Android system architecture
- Init – runtime – Zygoto
- Dalvik VM
- **Android specific kernel drivers**
- How to build Android

Linux kernel

- Many common Linux device drivers are available.
- Android specific kernel drivers
  - binder
  - ashmem
  - wake lock
  - logger
  - …
- http://elinux.org/Android_Kernel_Features
- These source code is not yet merged to kernel main line repository.

Binder

- /dev/binder
- Base of Inter Process Method Invocation
- Not for general purpose. Tuned for specific transaction.
- Multi-thread aware
  - Have internal data per thread
  - (CF. Socket have internal data per fd.)
- Doesn't use ”write” and ”read” system calls. Write and read at once by ”ioctl”.
- http://blog.kmckk.com/archives/3676340.html

Ashmem

- Android / Anonymous SHared MEMory subsystem
  - $(TOP)/system/core/cutils/ashmem.h
    - int ashmem_create_region(const char *name, size_t size) → returns fd
    - int ashmem_set_prot_region(int fd, int prot)
    - int ashmem_pin_region(int fd, size_t offset, size_t len)
    - int ashmem_unpin_region(int fd, size_t offset, size_t len)
- Kernel reclaims not ‘pin’ ed memory
- Similar to weak reference of Java. Useful to implement cache.
- android.os.MemoryFile from Java program

Wake lock

- Lock to prevent entering sleep mode.
- My memos
  - http://blog.kmckk.com/archives/3298375.html
  - http://blog.kmckk.com/archives/3304836.html
- eLinux wiki
  - http://elinux.org/Android_Power_Management

Alarm

- kernel implementation to support Android's AlarmManager.
- Wake up even when it was in sleep mode.

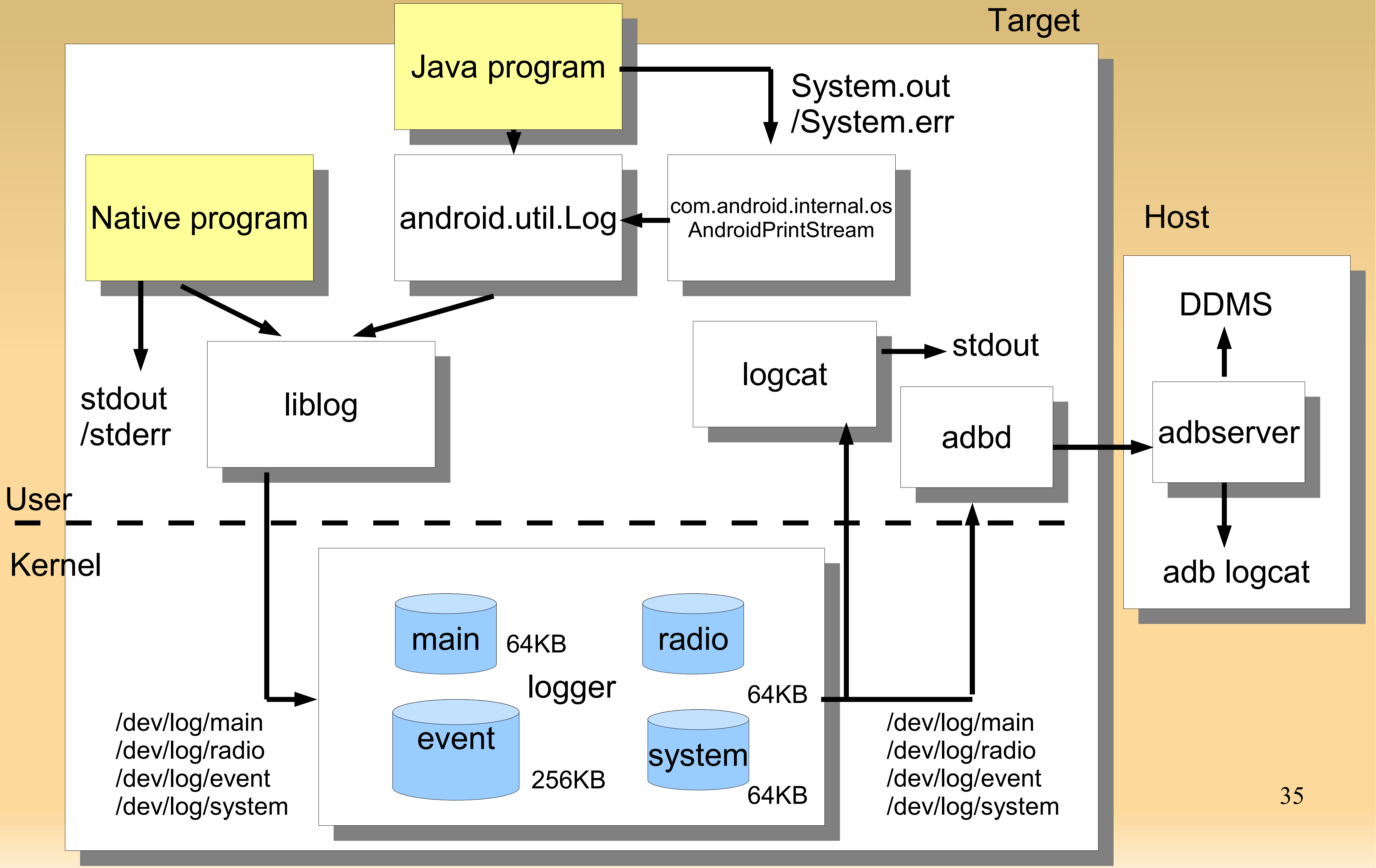
Low memory killer

- At the shortage of memory, the kernel select a process seems low priority and kill it. (!!)
- It's OK. because specification in the Android application life cycle, application should be preserve its own status.
- http://blog.kmckk.com/archives/2795577.html

Logger

- Android has unique system-wide log system
- http://blog.kmckk.com/archives/2936958.html
- http://elinux.org/Android_Logging_System

Overview of Android Logging System



Today's topic

- Android system architecture
- Init – runtime – Zygoto
- Dalvik VM
- Android specific kernel drivers
- **How to build Android**

How to build Android

- All source code is available for download
  - except Google specific services (Google map, Android market, …)
- Easy to download source and build them
- See AOSP web site
  - http://source.android.com/
- Or, my blog
  - http://blog.kmckk.com/archives/3722957.html

Conclusion

- Android system architecture is totally different from normal Linux systems.
- Android uses Linux kernel only, further more, adding android specific kernel drivers.
- Designed for Java applications.
- Tuned for small system.

Q & A

Thank you for listening!
Any comments to blogs are welcome.