KatDC
OPEN
SPACE
FORUM

0xlab

下一站， Android

Jim Huang (黃敬群 /jserv)

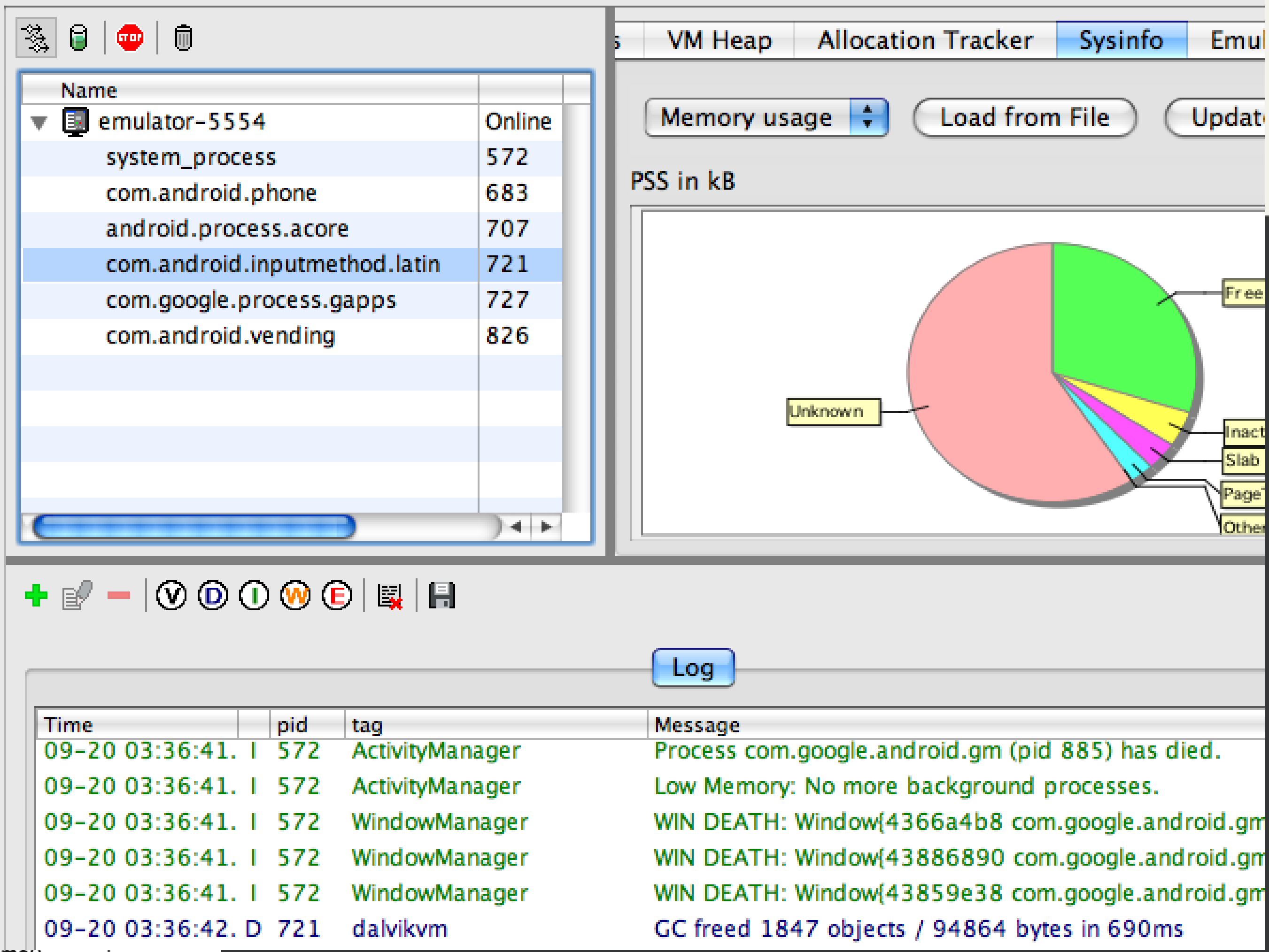
From 0xlab - http://0xlab.org/

Dec 12, 2009

- ✓「下一站」到底有多遠？
- ✓ 特立獨行的 **Android** ：相容性探討
- ✓ 系統效能分析與評估
- ✓ 功能層面的改進：以無線通訊爲例

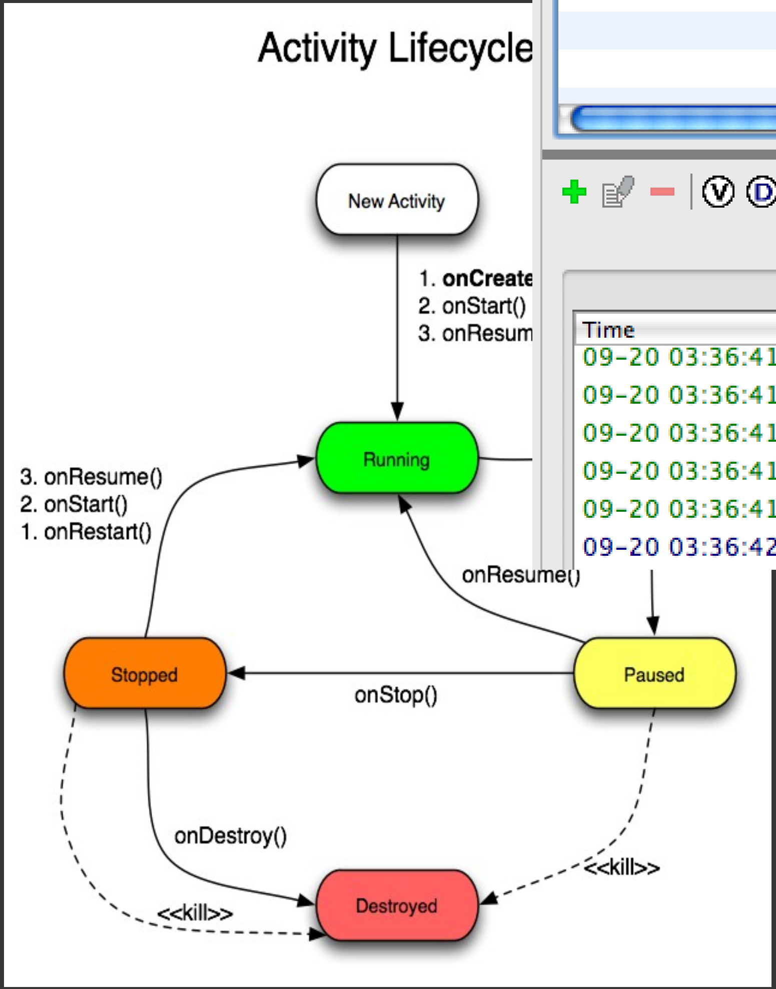
「下一站」到底有多遠？

```
 I/DEBUG ( 547): Build fingerprint: 'generic/sdk/generic/:1.5/CUPCAKE/148875:eng/test-keys'
I/DEBUG ( 547): pid: 732, tid: 746 >>> com.flexilis <<<
I/DEBUG ( 547): signal 11 (SIGSEGV), fault addr 00000002
I/DEBUG ( 547): r0 0000000b r1 00000000 r2 487ae000 r3 80000400
I/DEBUG ( 547): r4 00000002 r5 80000400 r6 001e6368 r7 00000000
I/DEBUG ( 547): r8 00000001 r9 4360d2f8 10 40008238 fp ad083e10
I/DEBUG ( 547): ip 00000002 sp 46319288 lr 00000004 pc ad01663c cpsr 20000010
I/DEBUG ( 547): #00 pc 0001663c /system/lib/libdvm.so
I/DEBUG ( 547): #01 pc 00016b78 /system/lib/libdvm.so
I/DEBUG ( 547): #02 pc 00014678 /system/lib/libdvm.so
I/DEBUG ( 547): #03 pc 00014798 /system/lib/libdvm.so
I/DEBUG ( 547): #04 pc 000148ac /system/lib/libdvm.so
I/DEBUG ( 547): #05 pc 00016bc0 /system/lib/libdvm.so
I/DEBUG ( 547): #06 pc 0001535c /system/lib/libdvm.so
I/DEBUG ( 547): #07 pc 00047c94 /system/lib/libdvm.so
I/DEBUG ( 547): #08 pc 00058a16 /system/lib/libdvm.so
I/DEBUG ( 547): #09 pc 00042dba /system/lib/libdvm.so
I/DEBUG ( 547): #10 pc 00029430 /system/lib/libdvm.so
I/DEBUG ( 547): #11 pc 00017610 /system/lib/libdvm.so
I/DEBUG ( 547): #12 pc 000520ec /system/lib/libdvm.so
```



Name	
emulator-5554	Online
system_process	572
com.android.phone	683
android.process.acore	707
com.android.inputmethod.latin	721
com.google.process.gapps	727
com.android.vending	826

```
Time               pid  tag              Message
09-20 03:36:41. I  572  ActivityManager  Process com.google.android.gm (pid 885) has died.
09-20 03:36:41. I  572  ActivityManager  Low Memory: No more background processes.
09-20 03:36:41. I  572  WindowManager    WIN DEATH: Window{4366a4b8 com.google.android.gm
09-20 03:36:41. I  572  WindowManager    WIN DEATH: Window{43886890 com.google.android.gm
09-20 03:36:41. I  572  WindowManager    WIN DEATH: Window{43859e38 com.google.android.gm
09-20 03:36:42. D  721  dalvikvm         GC freed 1847 objects / 94864 bytes in 690ms
```

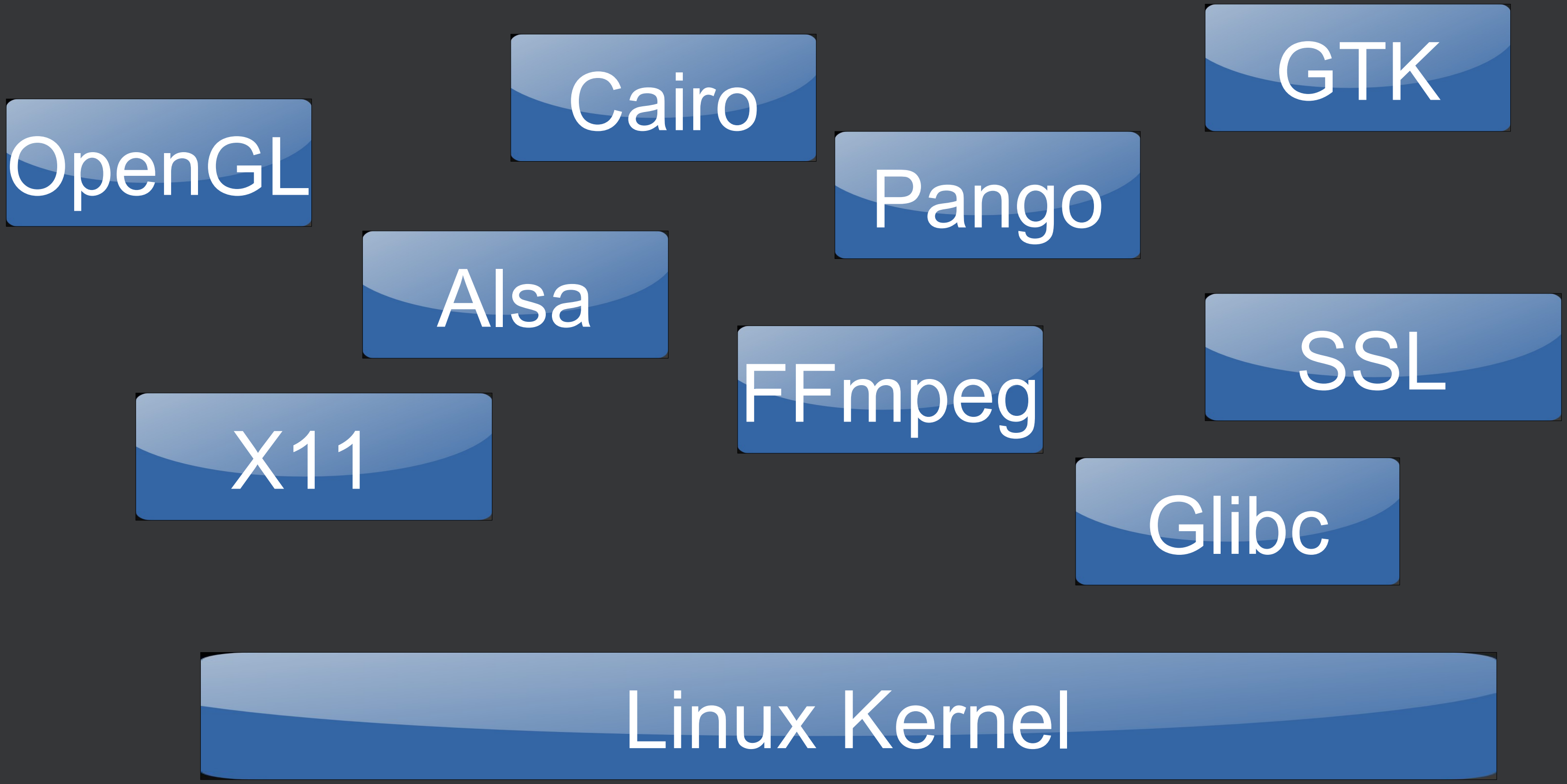


DDMS 是您的好朋友

Dalvik heap, GC, call stack, log, intent/activity

特立獨行的 Android

在 Android 中，您所熟悉的套件



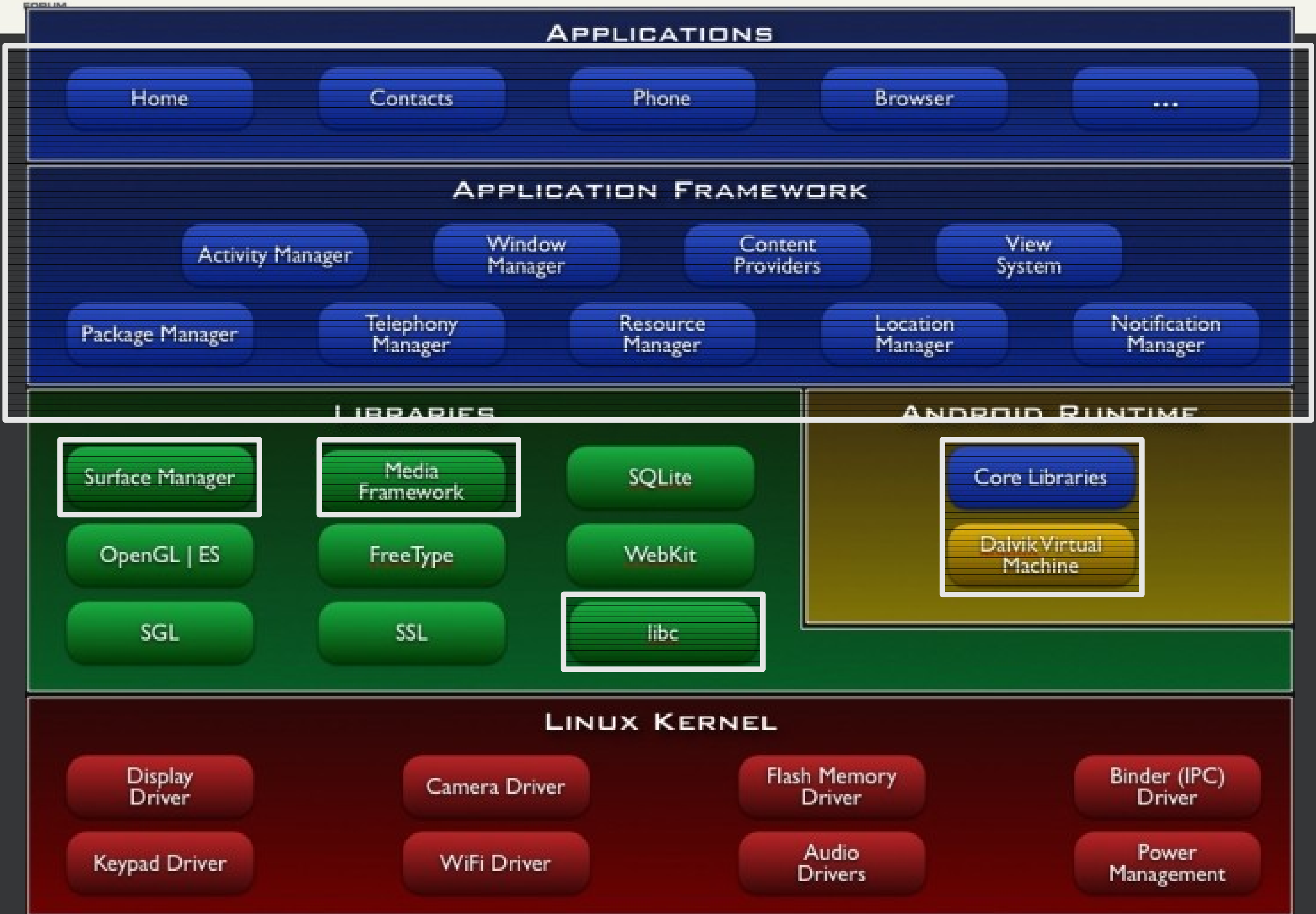
都被取代了

OpenGL

SSL

Linux Kernel

Android 的組成


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

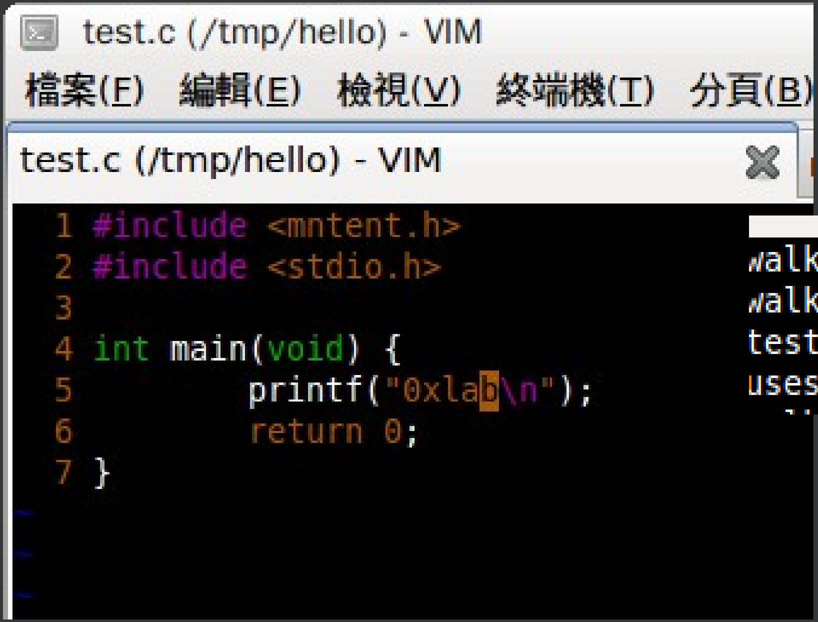
Android 砍掉重練

- glibc → bionic
- Cairo → Skia
- FFmpeg/GStreamer → opencore
- ld.so → linker
- Mesa3D/OpenGL

  → PixelFlinger + OpenGL|ES
- init
- (Dalvik)

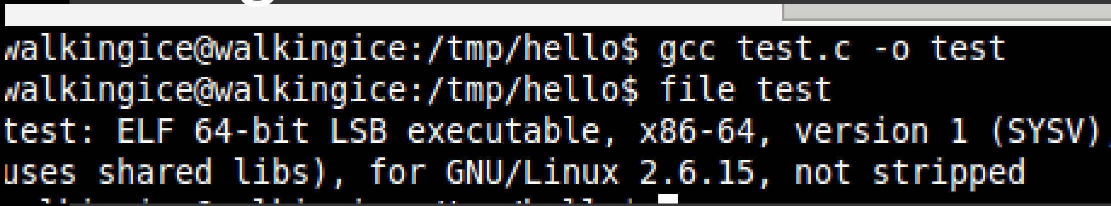
砍掉重練不是壞事
只是重新長出來的東西很不一樣

libc 大不同


test.c (/tmp/hello) - VIM
檔案(F) 編輯(E) 檢視(V) 終端機(T) 分頁(B)
test.c (/tmp/hello) - VIM

```c
#include <mntent.h>
#include <stdio.h>

int main(void) {
        printf("0xlab\n");
        return 0;
}
```

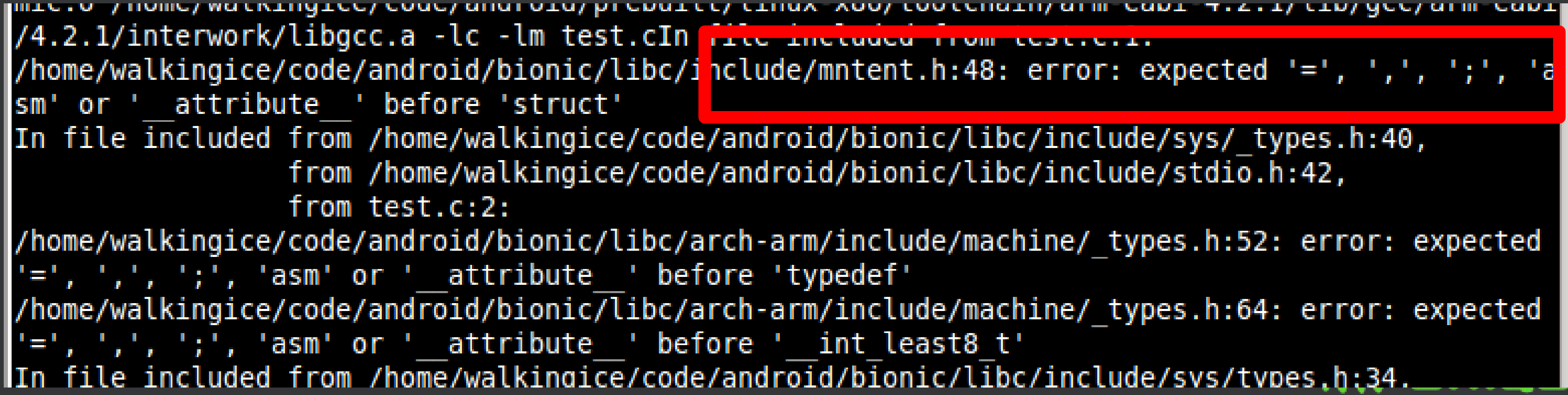
glibc - OK!



```
walkingice@walkingice:/tmp/hello$ gcc test.c -o test
walkingice@walkingice:/tmp/hello$ file test
test: ELF 64-bit LSB executable, x86-64, version 1 (SYSV)
uses shared libs), for GNU/Linux 2.6.15, not stripped
```

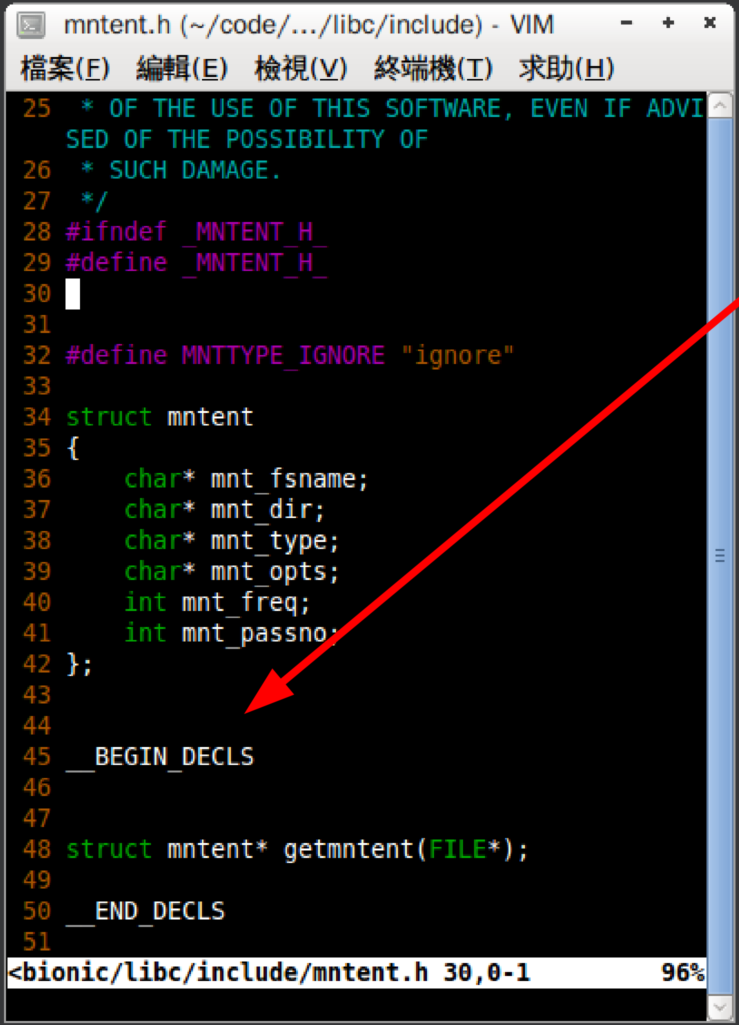
bionic：一連串的錯誤

(非 POSIX 相容)



```
/4.2.1/interwork/libgcc.a -lc -lm test.cIn file included from test.c:1:
/home/walkingice/code/android/bionic/libc/include/mntent.h:48: error: expected '=', ',', ';', 'a
sm' or '__attribute__' before 'struct'
In file included from /home/walkingice/code/android/bionic/libc/include/sys/_types.h:40,
                 from /home/walkingice/code/android/bionic/libc/include/stdio.h:42,
                 from test.c:2:
/home/walkingice/code/android/bionic/libc/arch-arm/include/machine/_types.h:52: error: expected
'=', ',', ';', 'asm' or '__attribute__' before 'typedef'
/home/walkingice/code/android/bionic/libc/arch-arm/include/machine/_types.h:64: error: expected
'=', ',', ';', 'asm' or '__attribute__' before '__int_least8_t'
In file included from /home/walkingice/code/android/bionic/libc/include/sys/types.h:34,
```

libc 大不同


mntent.h (~/code/.../libc/include) - VIM
檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)

```
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
#ifndef _MNTENT_H_
#define _MNTENT_H_


#define MNTTYPE_IGNORE "ignore"

struct mntent
{
    char* mnt_fsname;
    char* mnt_dir;
    char* mnt_type;
    char* mnt_opts;
    int mnt_freq;
    int mnt_passno;
};


__BEGIN_DECLS


struct mntent* getmntent(FILE*);

__END_DECLS

<bionic/libc/include/mntent.h 30,0-1        96%
```

在 sys/cdefs.h 定義

gcc:
mntent.h → features.h → cdefs.h

bionic:
mntent → ???

兩邊的實作方式不一樣

libc 大不同

Iptables, wireless-tools

glibc

```
        #include <net/ethernet.h>
```

bionic

```
        #include <net/if_net.h>
```

glibc

```
        #include <sys/soundcard.h>
```

bionic

```
        #include <linux/soundcard.h>
```

ipv6

glibc: /usr/include/netinet/in.h

bionic: ???

沒有 getline, dprintf, vdprintf...etc

Very limited support for SysV, POSIX

/usr/include/bits/socket.h

```
/* Type for length arguments in socket calls.  */
#ifndef __socklen_t_defined
typedef __socklen_t socklen_t;
# define __socklen_t_defined
#endif
```

bionic/libc/include/sys/socket.h

```
typedef int socklen_t;
```

這些都算小問題，雖然會增加移植的成本
終究還是能夠解決

(bionic) 沒有 IPC 的 shared memory

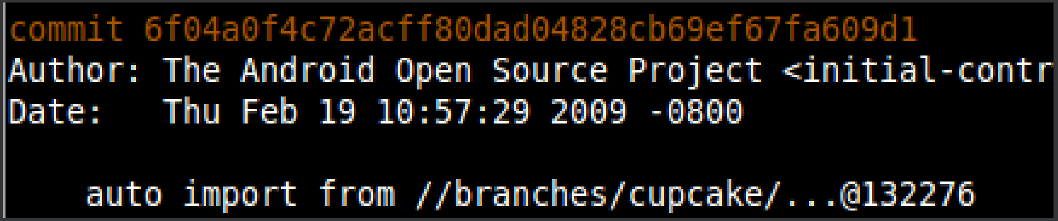
bionic/libc/docs/SYSV-IPC.TXT

Android does not support System V IPCs, i.e. the facilities provided by the following standard Posix headers:

```
  <sys/sem.h>   /* SysV semaphores */
  <sys/shm.h>   /* SysV shared memory segments */
  <sys/msg.h>   /* SysV message queues */
  <sys/ipc.h>   /* General IPC definitions */
```

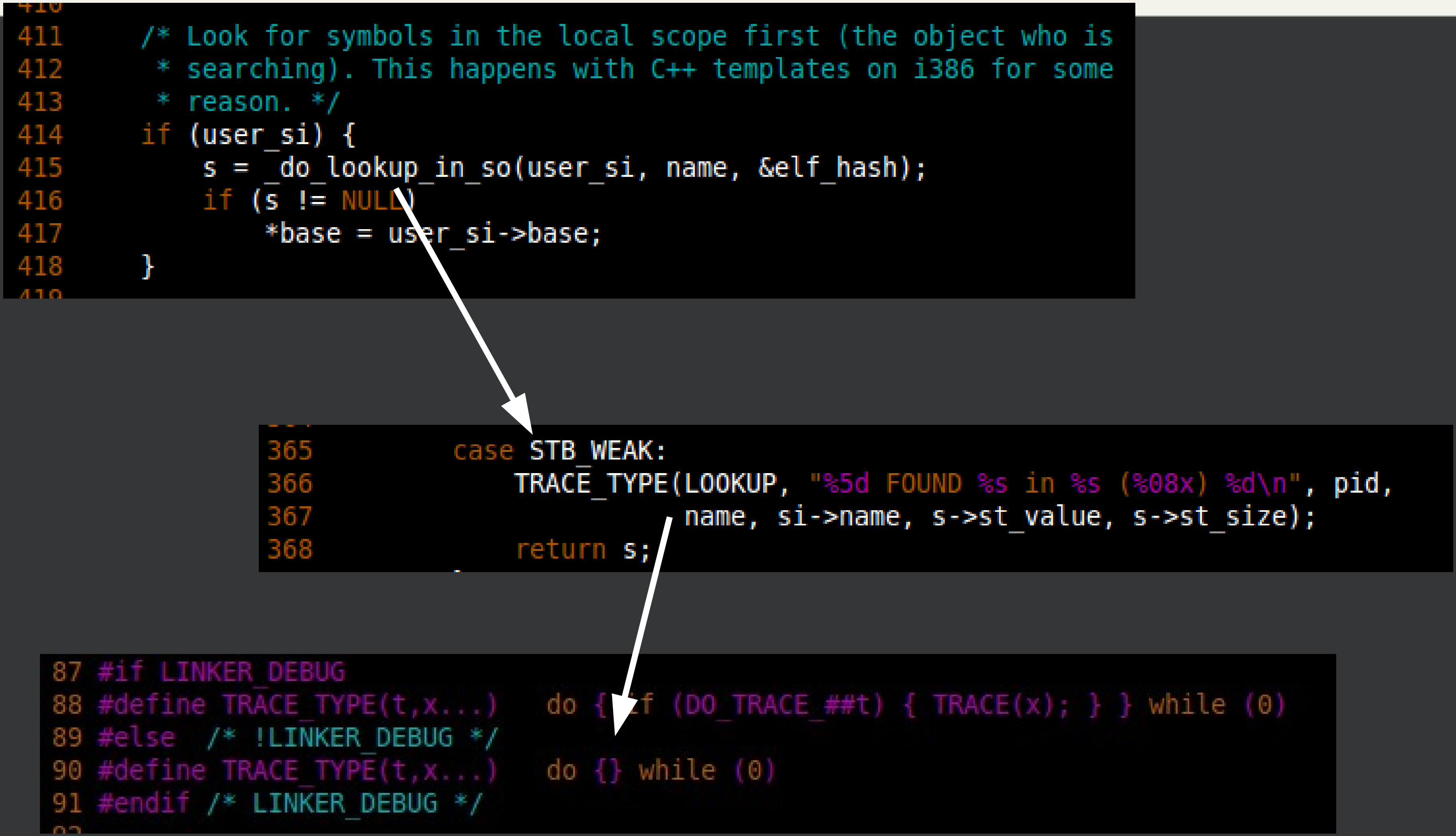
The reason for this is due to the fact that, by design, they lead to global kernel resource leakage.

這份文件第一次出現在 2009/02/19



```
commit 6f04a0f4c72acff80dad04828cb69ef67fa609d1
Author: The Android Open Source Project <initial-contr
Date:   Thu Feb 19 10:57:29 2009 -0800

    auto import from //branches/cupcake/...@132276
```

(bionic)**linker/linker.c**



```
    /* Look for symbols in the local scope first (the object who is
     * searching). This happens with C++ templates on i386 for some
     * reason. */
    if (user_si) {
        s = _do_lookup_in_so(user_si, name, &elf_hash);
        if (s != NULL)
            *base = user_si->base;
    }
```

```
            case STB_WEAK:
                TRACE_TYPE(LOOKUP, "%5d FOUND %s in %s (%08x) %d\n", pid,
                           name, si->name, s->st_value, s->st_size);
                return s;
```

```
#if LINKER_DEBUG
#define TRACE_TYPE(t,x...)   do { if (DO_TRACE_##t) { TRACE(x); } } while (0)
#else  /* !LINKER_DEBUG */
#define TRACE_TYPE(t,x...)   do {} while (0)
#endif /* LINKER_DEBUG */
```

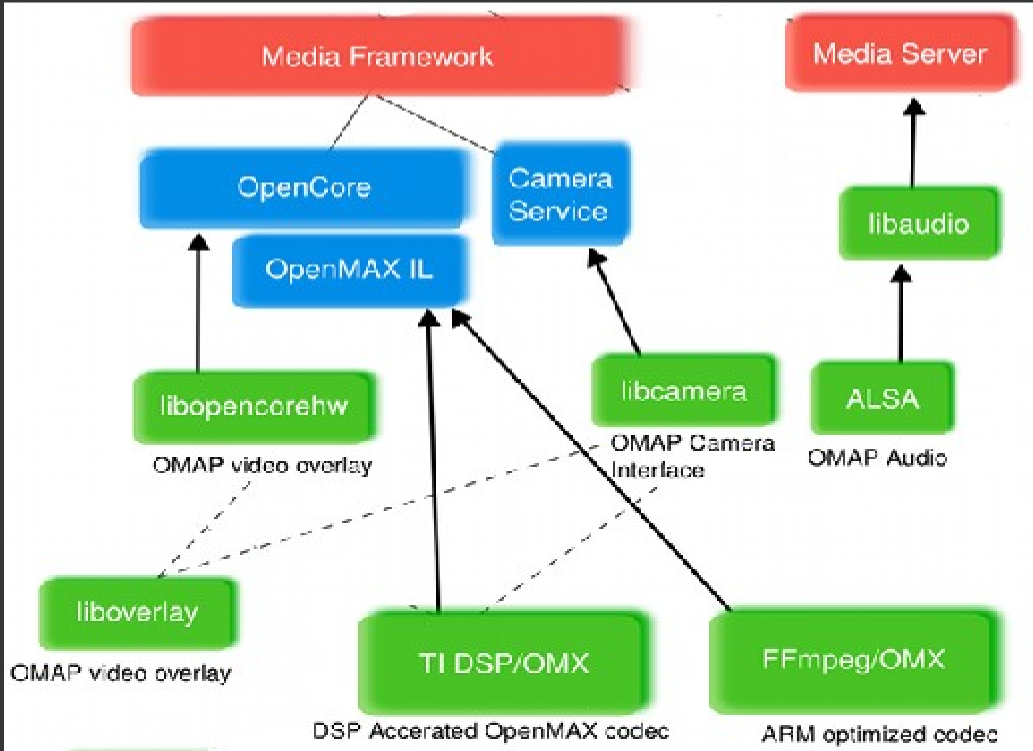
Linker 的設計沒有處理 weak symbol
遇到 weak symbol 便傳回錯誤的值

系統效能分析與評估

SoC specific module (minimal efforts)

- **libopencorehw.so** (OpenCore HW module)
  http://gitorious.org/0xdroid/hardware_omap3_libopencorehw
- **liboverlay.so** (Graphics overlays module)
- **libcamera.so** (Camera HAL)
  http://gitorious.org/0xdroid/hardware_omap3_camera
- **libaudio.so** (Audio HAL)
  http://gitorious.org/0xdroid/hardware_alsa_sound

0xdroid provides the full source code of reference hardware acceleration modules for Android.

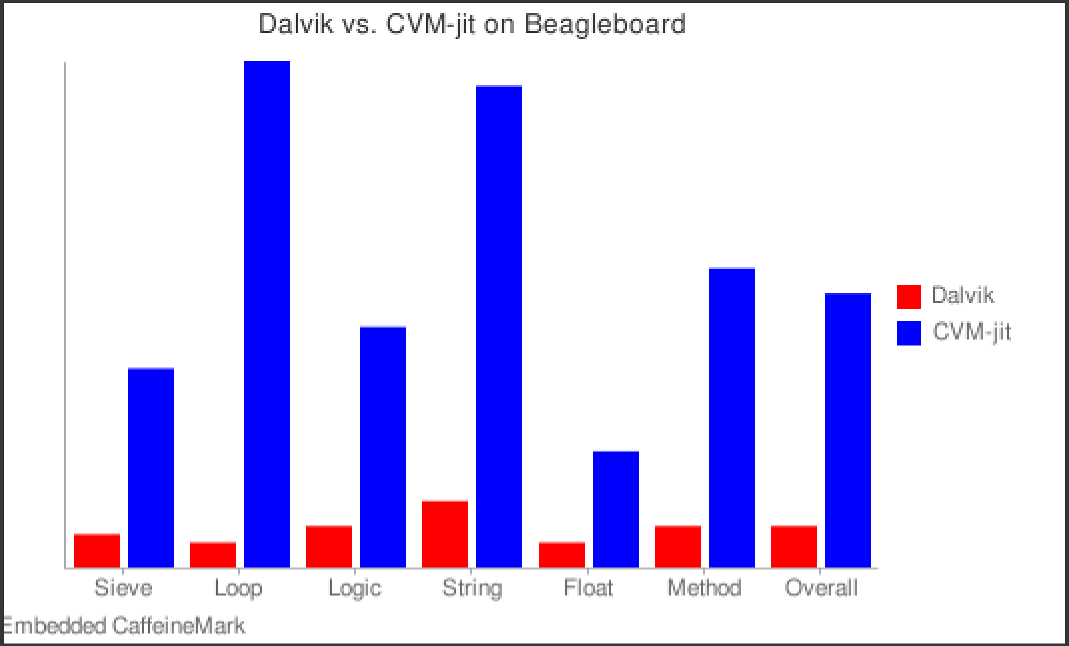


Performance Evaluation on Beagleboard

- **TI OMAP3 SoC powered**
- **500 MHz / ARM Cortex A8**
- **0xdroid – well-tuned Android for Beagleboard (TI OMAP 3530)**
  - **http://code.google.com/p/0xdroid/**
- **Based on Android Donut branch**
  - beagle-cupcake-0x2, beagle-donut-0x3
- Expertise in Android **porting** and performance **tuning**.

Dalvik VM

- Embedded CaffeineMark
- **Dalvik VM : 1034**
- **CVM + JIT : 7526**



- Dalvik + bionic : CVM pure interpreter + glibc
  - Sieve → 950 : 351
  - Loop → 775 : 329
  - Logic → 1104 : 286
  - String → 1898 : 3023
  - Float → 772 : 298
  - Method → 1032 : 286

Dalvik VM

Refernce CaffeineMarkEmbedded results: (OMAP3530 at 500MHz)

[[eclair + armv7 interpreter]]	[[eclair + armv7 + jit]]
Sieve score = 956 (98)	Sieve score = 2345 (98)
Loop score = 783 (2017)	Loop score = 3629 (2017)
Logic score = 1099 (0)	Logic score = 5618 (0)
String score = 2019 (708)	String score = 4328 (708)
Float score = 819 (185)	Float score = 1495 (185)
Method score = 1103 (166650)	Method score = 1954 (166650)
Overall score = 1069	**Overall score = 2907**

但 JIT compiler 不是萬靈丹，

[x] 更大的記憶體開銷 (x2)

[x] 較慢的應用程式啟動時間

[x] GC 的非預期表現

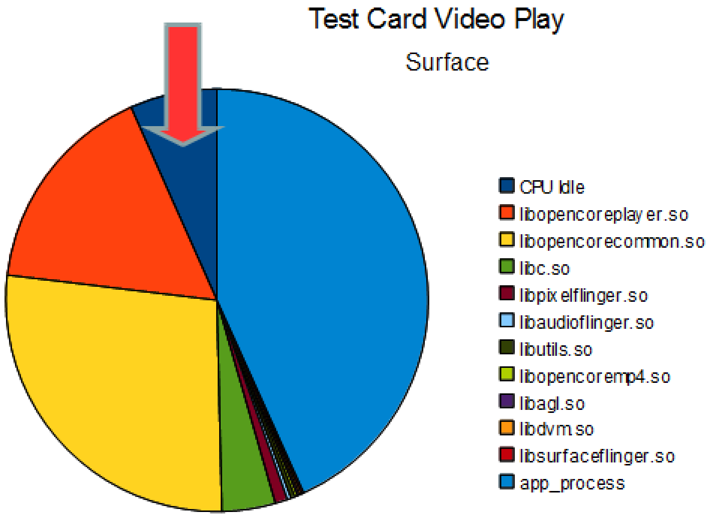
Evalutions scenario: Introduced libopencorehw.so

(measured by utility "oprofile")

Video playback :: Test Card Video (480x360, 25fps, H.264)

CPU Idle	6.65
libopencoreplayer.so	16.35
libopencorecommon.so	27.15
libc.so	4.08
libpixelflinger.so	0.93
libaudioflinger.so	0.3
libutils.so	0.3
libopencoremp4.so	0.22
libagl.so	0.18
libdvm.so	0.17
libsurfaceflinger.so	0.16
app_process	43.06

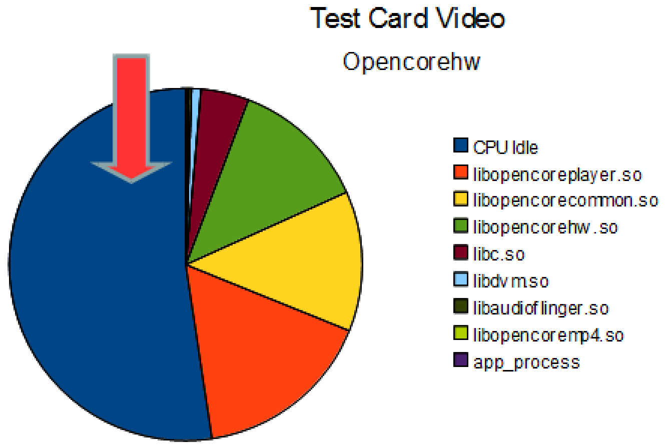
Action video play (surface, original)



Idle: **6.65%** vs. **51.96%**
Reduce system computing power by introducing hardware overlay

CPU Idle	51.96
libopencoreplayer.so	16.36
libopencorecommon.so	12.78
libopencorehw.so	12.39
libc.so	4.35
libdvm.so	0.84
libaudioflinger.so	0.26
libopencoremp4.so	0.21
app_process	0.05

Action video play (overlay, 0xlab)



Evalutions scenario: Introduced libopencorehw.so

(measured by utility “oprofile”)

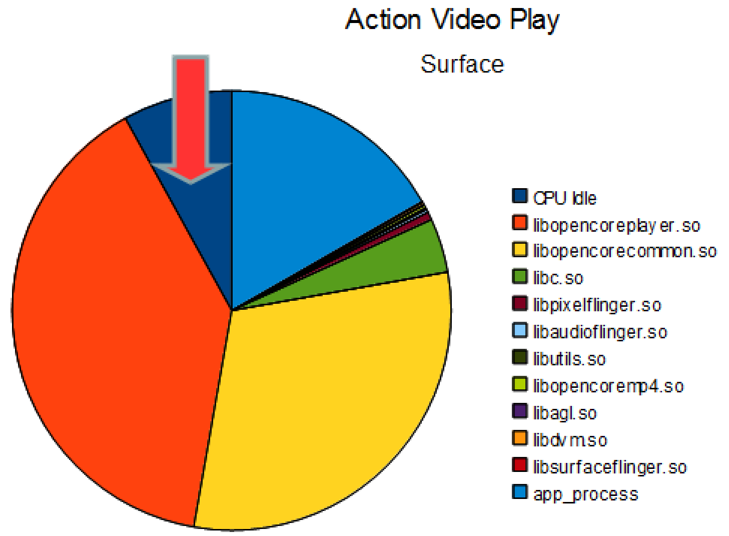
Video playback :: Action Video (480x360, 25fps, H.264)

Action video play (surface, original)

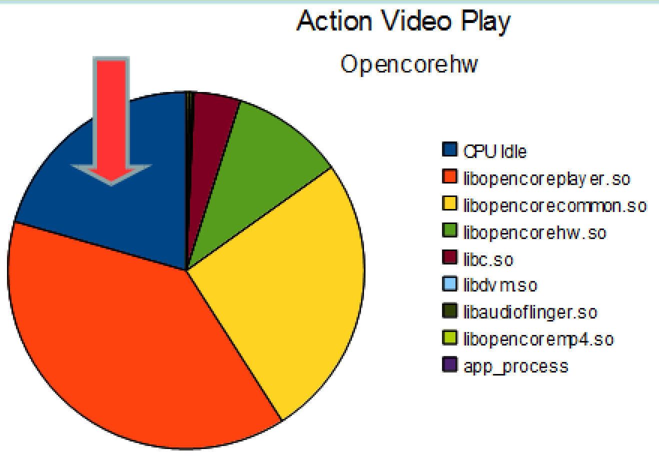
CPU Idle	8.03
libopencoreplayer.so	39.05
libopencorecommon.so	30.47
libc.so	3.97
libpixelflinger.so	0.57
libaudioflinger.so	0.23
libutils.so	0.14
libopencoremp4.so	0.23
libagl.so	0.04
libdvm.so	0.19
libsurfaceflinger.so	0.06
app_process	16.68

Action video play (overlay, 0xlab)

CPU Idle	20.49
libopencoreplayer.so	38.32
libopencorecommon.so	25.69
libopencorehw.so	10.2
libc.so	4.19
libdvm.so	0.17
libaudioflinger.so	0.28
libopencoremp4.so	0.23
app_process	0.03



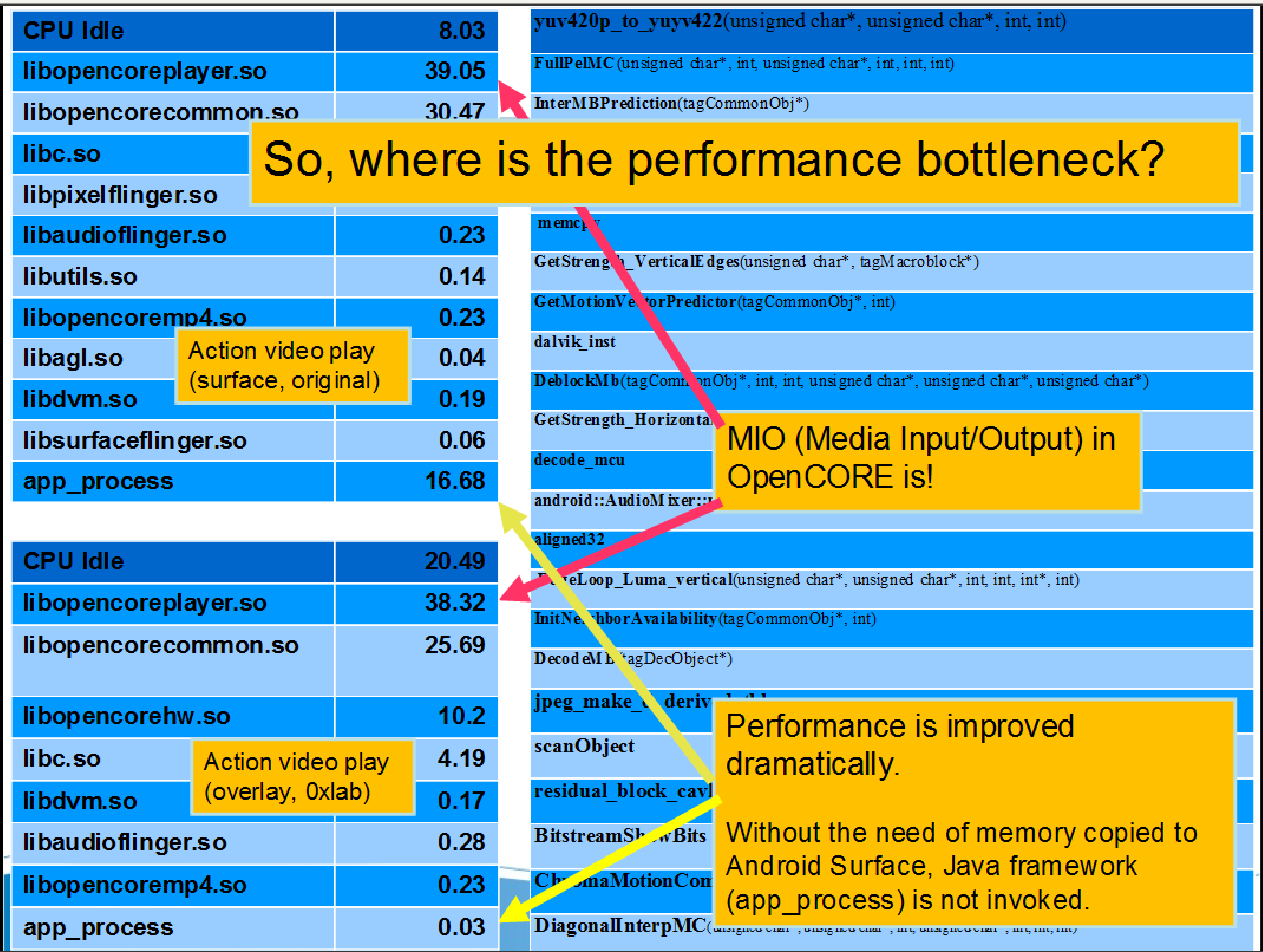
Idle: **8.03%** vs. **20.49%**
Even codec is quite busy, system computing power benefits from hardware overlays.



Evalutions scenario: Introduced libopencorehw.so

(measured by utility “oprofile”)

Video playback :: Action Video (480x360, 25fps, H.264)



Action video play (surface, original)

CPU Idle	8.03
libopencoreplayer.so	39.05
libopencorecommon.so	30.47
libc.so	
libpixelflinger.so	
libaudioflinger.so	0.23
libutils.so	0.14
libopencoremp4.so	0.23
libagl.so	0.04
libdvm.so	0.19
libsurfaceflinger.so	0.06
app_process	16.68

Action video play (overlay, 0xlab)

CPU Idle	20.49
libopencoreplayer.so	38.32
libopencorecommon.so	25.69
libopencorehw.so	10.2
libc.so	4.19
libdvm.so	0.17
libaudioflinger.so	0.28
libopencoremp4.so	0.23
app_process	0.03

yuv420p_to_yuyv422(unsigned char*, unsigned char*, int, int)
FullPelMC(unsigned char*, int, unsigned char*, int, int, int)
InterMBPrediction(tagCommonObj*)
memcpy
GetStrength_VerticalEdges(unsigned char*, tagMacroblock*)
GetMotionVectorPredictor(tagCommonObj*, int)
dalvik_inst
DeblockMb(tagCommonObj*, int, int, unsigned char*, unsigned char*, unsigned char*)
GetStrength_Horizonta
decode_mcu
android::AudioMixer::
aligned32
eLoop_Luma_vertical(unsigned char*, unsigned char*, int, int, int*, int)
InitNeighborAvailability(tagCommonObj*, int)
DecodeMB(tagDecObject*)
jpeg_make_ deriv
scanObject
residual_block_cavl
BitstreamShowBits
ChromaMotionCom
DiagonalInterpMC(

So, where is the performance bottleneck?

MIO (Media Input/Output) in OpenCORE is!

Performance is improved dramatically.

Without the need of memory copied to Android Surface, Java framework (app_process) is not invoked.

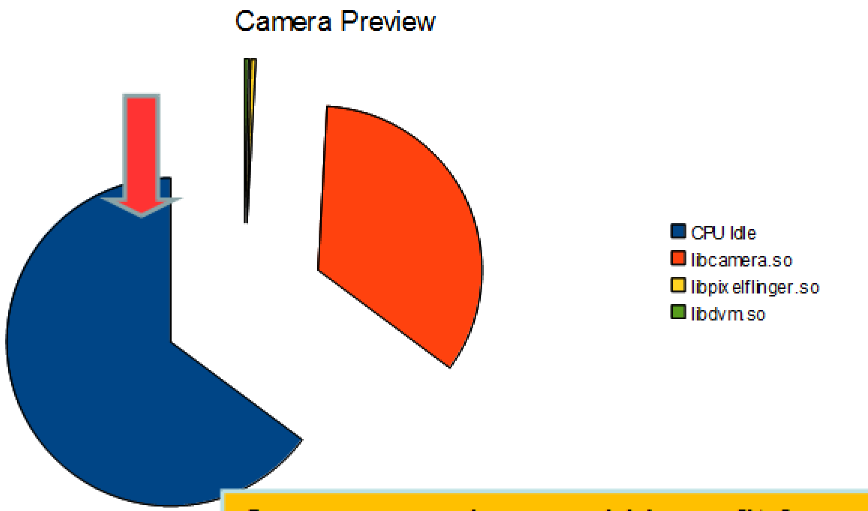
Evalutions scenario: Introduced libcamera.so

(measured by utility “oprofile”)

Camera preview (320x480)

CPU Idle	61.88
libcamera.so	32.73
libpixelflinger.so	0.47
libdvm.so	0.35

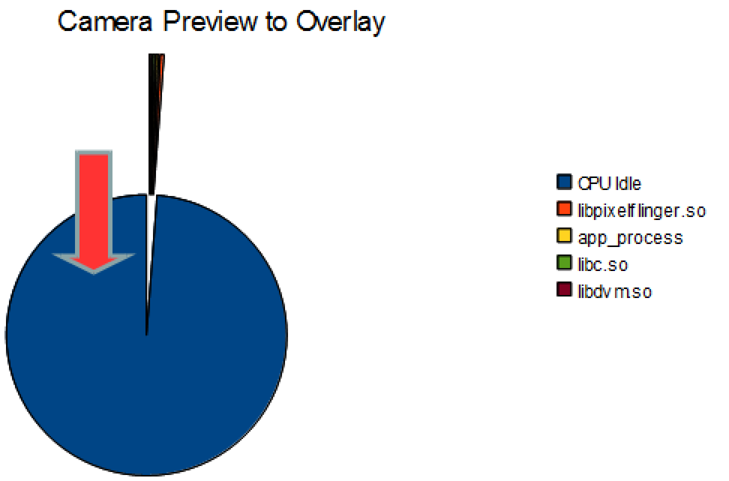
Action video play (surface, old)



Idle: **61.88%** vs. **98.38%**

Camera is quite important in Android,especially for rich applications such as bar-code / QR code scanner. These camera related applications usually requires preview screen.

Camera preview could benefit from the experience of video playback + hardware overlays



Action video play (overlay, 0xlab)

CPU Idle	98.38
libpixelflinger.so	0.44
app_process	0.28
libc.so	0.26
libdvm.so	0.23

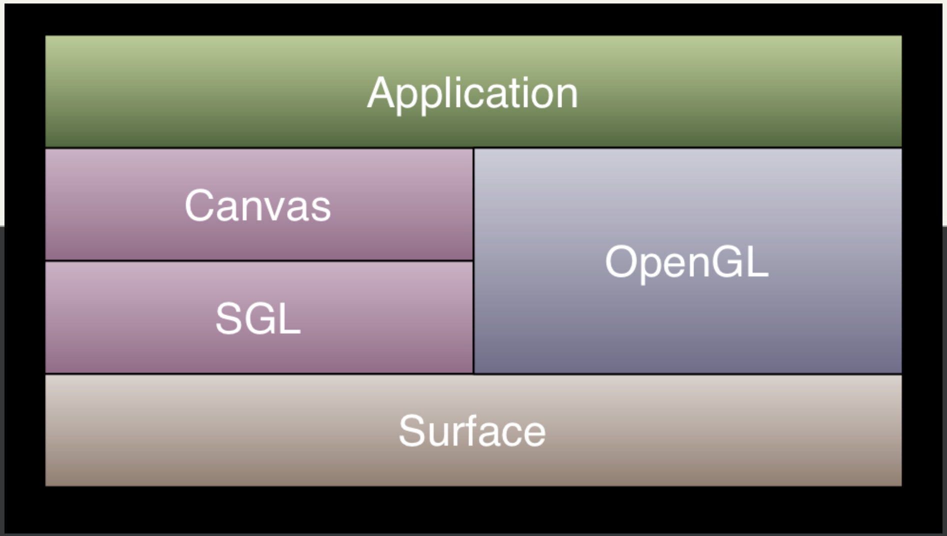
Memory operation tweaks

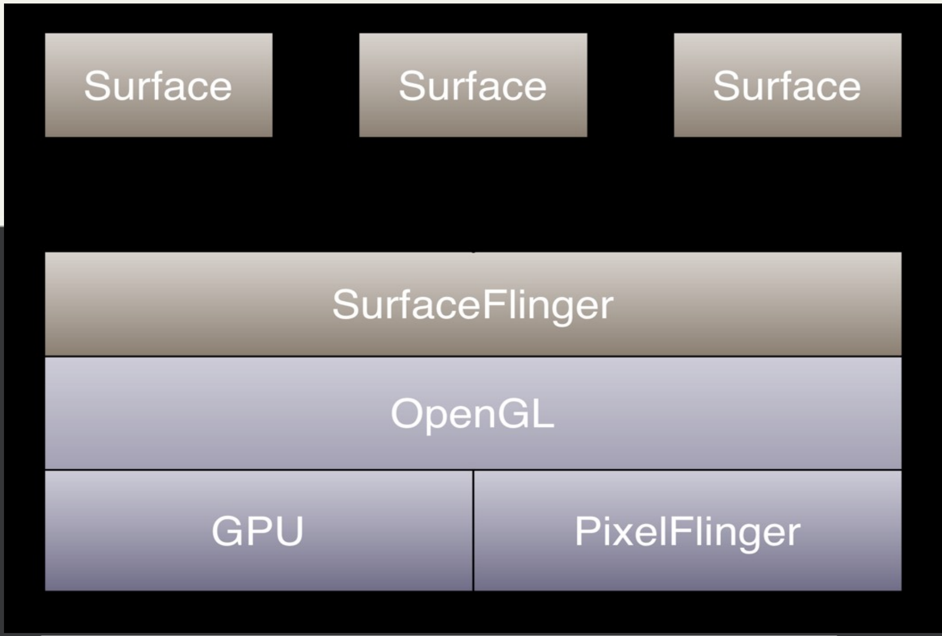
Dalvik, PixelFlinger, Skia

```
[[ very small data test ]]
memcpy_neon3 :  (24 bytes copy) =  152.2 MB/s /  312.2 MB/s
memcpy_neon2 :  (24 bytes copy) =  230.9 MB/s /  551.4 MB/s
memcpy_neon :  (24 bytes copy) =  198.7 MB/s /  349.3 MB/s
memcpy_armv5 :  (24 bytes copy) =  123.3 MB/s /  252.8 MB/s
memcpy_arm  :  (24 bytes copy) =  170.2 MB/s /  226.6 MB/s

memcpy_neon3 :  (31 bytes copy) =  201.8 MB/s /  218.4 MB/s
memcpy_neon2 :  (31 bytes copy) =  314.9 MB/s /  712.5 MB/s
memcpy_neon :  (31 bytes copy) =  267.4 MB/s /  374.7 MB/s
memcpy_armv5 :  (31 bytes copy) =  143.2 MB/s /  326.6 MB/s
memcpy_arm  :  (31 bytes copy) =  197.0 MB/s /  272.1 MB/s
```

```
[[ L1 cached data ]]
memcpy_neon3 :  (4096 bytes copy) = 1962.4 MB/s / 1910.9 MB/s
memcpy_neon2 :  (4096 bytes copy) = 2132.7 MB/s / 2192.7 MB/s
memcpy_neon :  (4096 bytes copy) = 2080.5 MB/s / 2230.7 MB/s
memcpy_armv5 :  (4096 bytes copy) =  806.8 MB/s / 1289.2 MB/s
memcpy_arm  :  (4096 bytes copy) =  830.5 MB/s / 1396.0 MB/s

memcpy_neon3 :  (6144 bytes copy) = 2006.7 MB/s / 1935.0 MB/s
memcpy_neon2 :  (6144 bytes copy) = 2176.2 MB/s / 2216.7 MB/s
memcpy_neon :  (6144 bytes copy) = 2139.9 MB/s / 2238.1 MB/s
memcpy_armv5 :  (6144 bytes copy) =  820.0 MB/s / 1300.5 MB/s
memcpy_arm  :  (6144 bytes copy) =  839.8 MB/s / 1411.7 MB/s
```

- memcpy_neon3 : Eclair's NEON optimized
- memcpy_neon2 : LGPL'd NEON optimized (version 2)
- memcpy_neon : LGPL'd NEON optimized
- memcpy_armv5 : donut/cupcake ARMv5 optimized
- memcpy_arm : LGPL ARMv5 optimized





PixelFlinger JIT

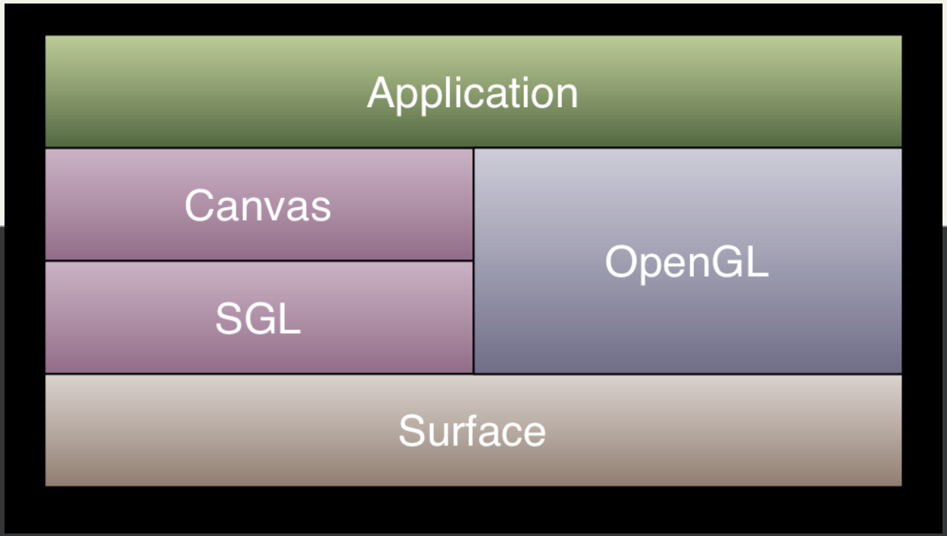
NEON instructions
Advanced ARM SIMD

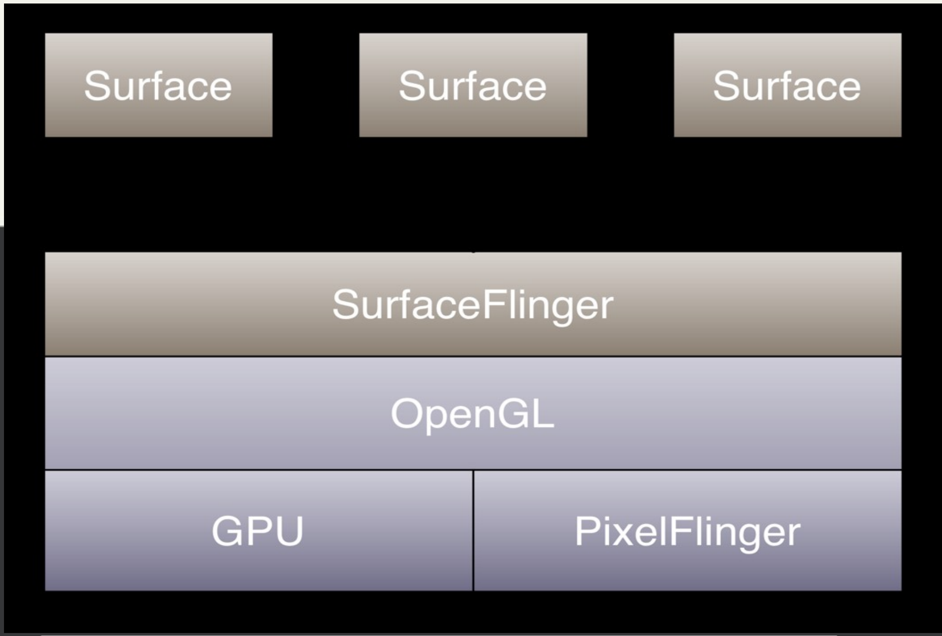
optimized scanline_t32cb16

Reference benchmark on Beagleboard (TI OMAP353x) at 500 MHz

```
    scanline_t32cb16_c memory bandwidth:        31.63 MB/s
    scanline_t32cb16_neon memory bandwidth: 147.69 MB/s
```

It could dramatically improve boot animation performance.





PixelFlinger JIT

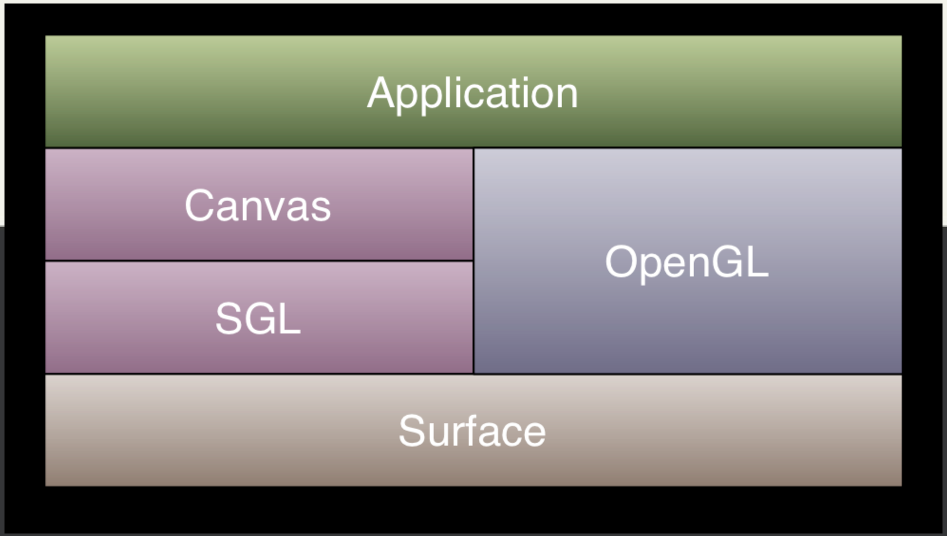
NEON instructions
Advanced ARM SIMD

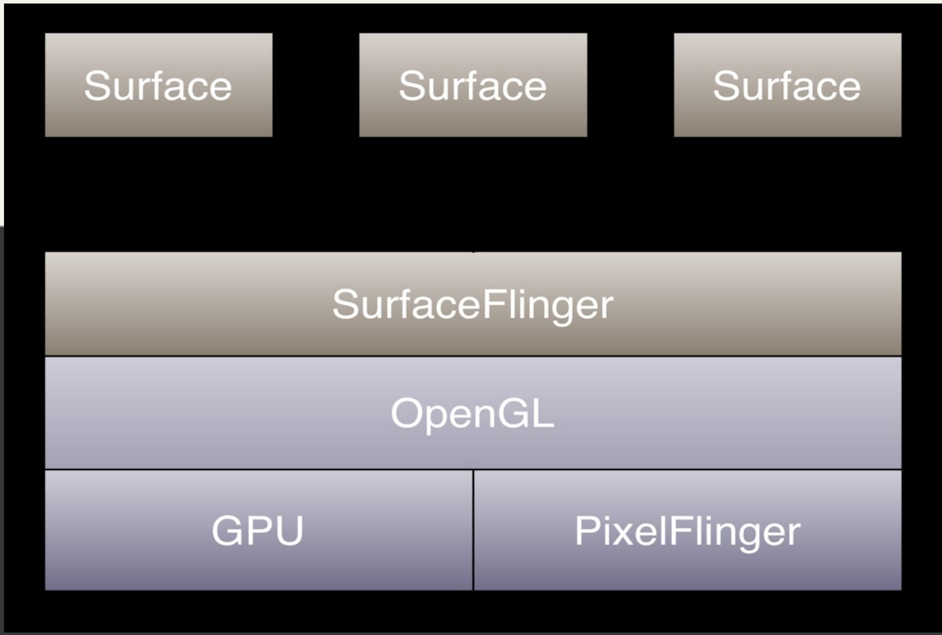
optimized t32cb16blend

Reference benchmark on Beagleboard 500MHz:

```
    scanline_t32cb16blend_c memory bandwidth:        12.81 MB/s
    scanline_t32cb16blend_arm memory bandwidth:      57.61 MB/s
    scanline_t32cb16blend_neon memory bandwidth: 128.66 MB/s
```

scanline_t32cb16blend_c: generic C implementation.
scanline_t32cb16blend_arm: ARMv5 optimized by Android.
scanline_t32cb16blend_neon: ARMv7 tweaked implementation.





UBFX instruction

Signed and Unsigned Bit Field Extract. Copies adjacent bits from one register into the least significant bits of a second register, and sign extends or zero extends to 32 bits.

PixelFlinger JIT

00000077:03515104_00000000_00000000

(Blends a single color into an RGB565 buffer.)

Before: 27 inst/pixel, After: 24 inst/pixel, Improvement: 12.5%

00000077:03545404_00000A01_00000000

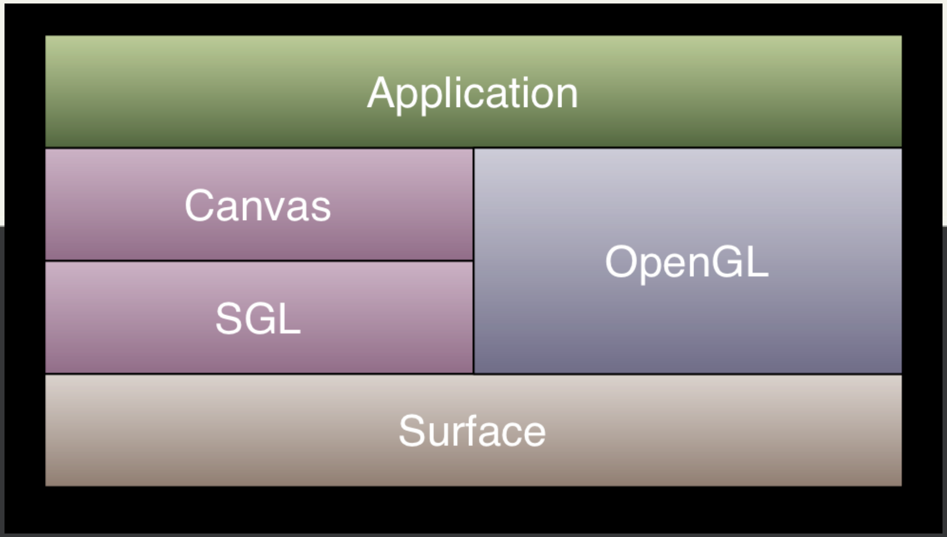
(Blends RGBA8888 texture into an RGB565 buffer using alpha.)

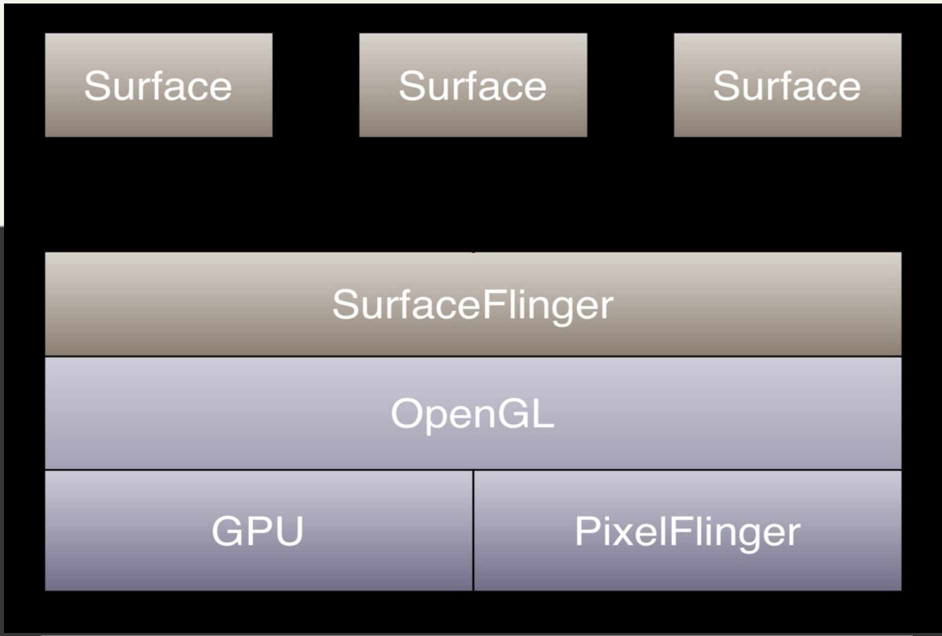
Before: 30 inst/pixel, After: 27 inst/pixel, Improvement: 11.1%

00000077:03545404_00000A04_00000000

(Blends RGB565 texture into an RGB565 buffer using alpha.)

Before: 29 inst/pixel, After: 27 inst/pixel, Improvement: 7.4%





UBXTB16 instruction

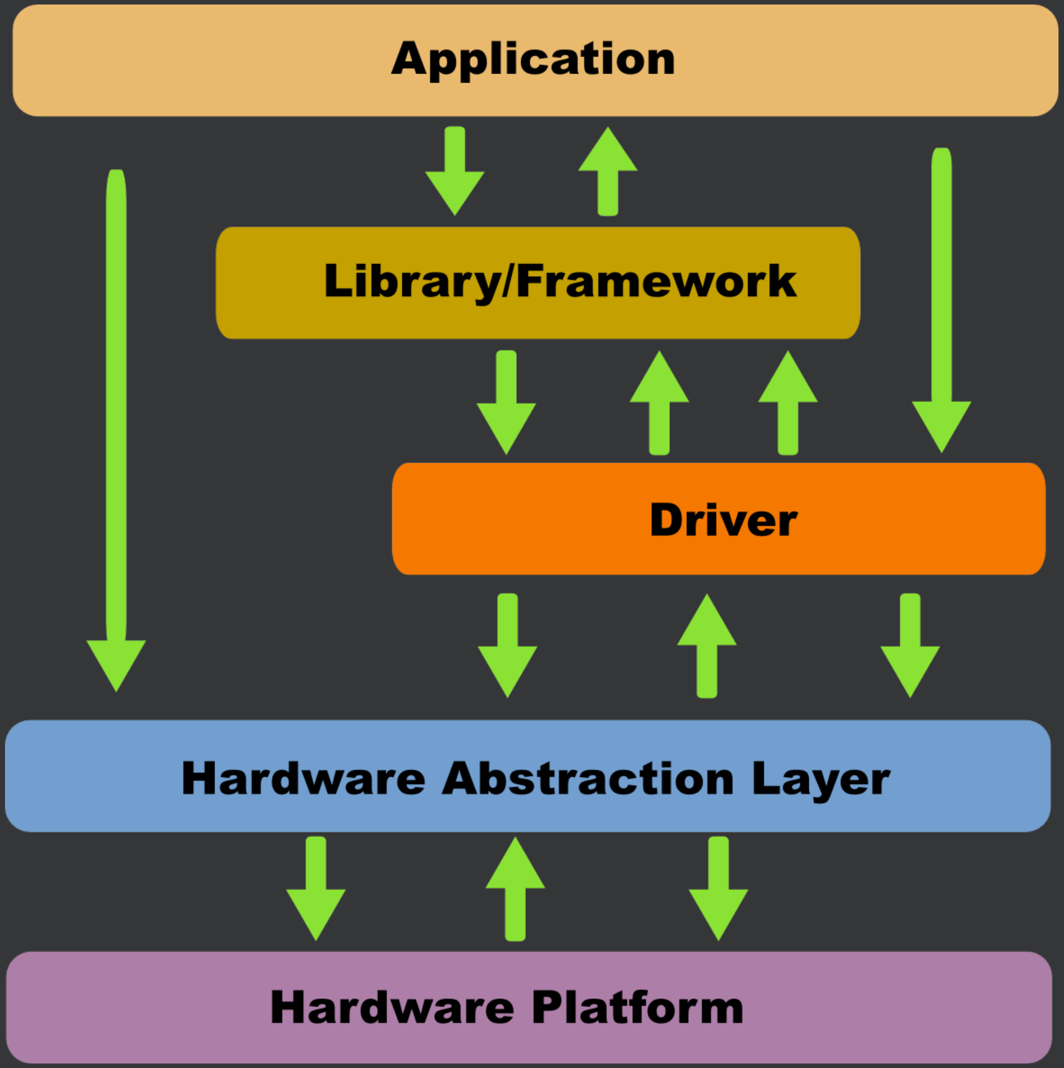
Introducing the UXTB16 instruction allows removal of some masking code, and is beneficial from a pipeline point of view - lots of UXTB16 followed by MUL sequences.

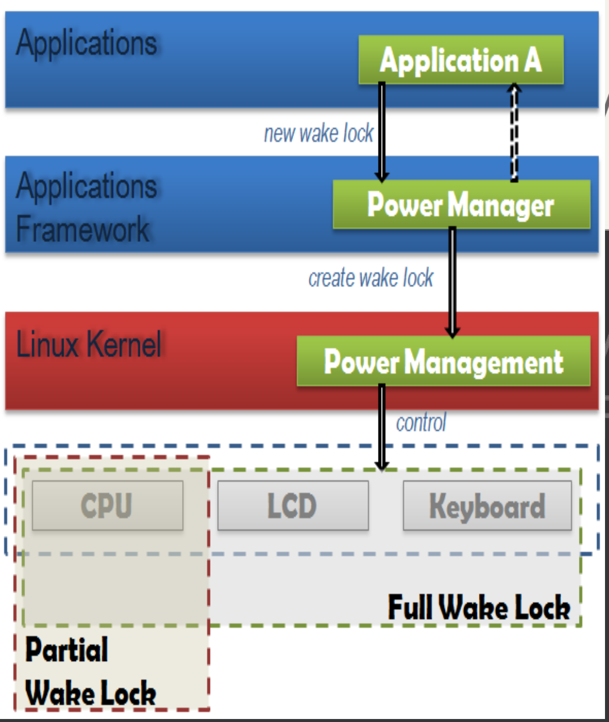
PixelFlinger JIT

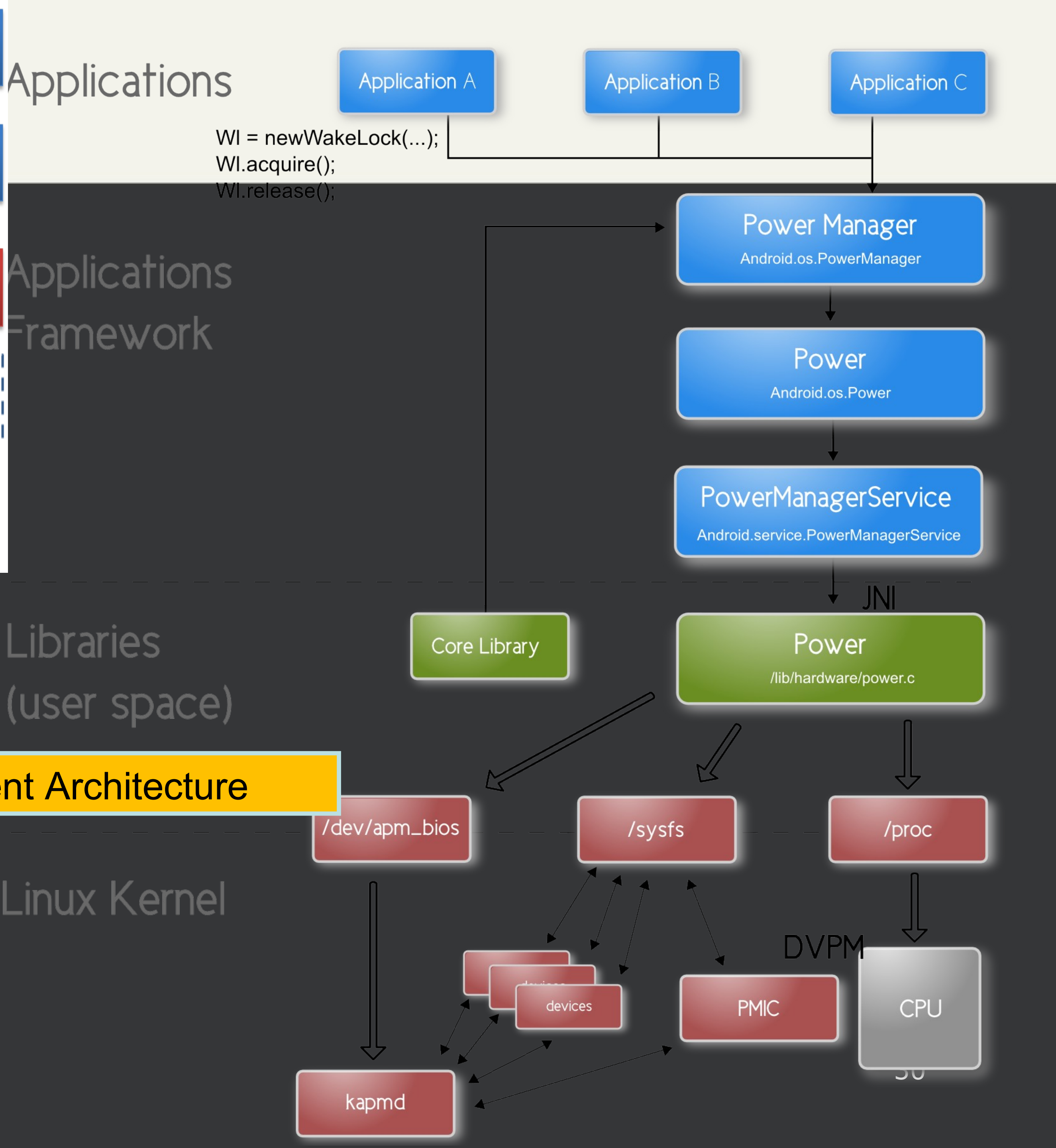
Code has been scheduled for A8 pipeline, specifically aiming to allow multiplies to issue in pipeline 0, for efficient dual issue operation.

Testing on SpriteMethodTest (http://code.google.com/p/apps-for-android/) gives 8% improvement (12.7 vs. 13.7 fps.)

功能層面的改進： Wireless







Android Power Management Architecture

KatDC
OPEN
SPACE
FORUM

Thank you !

Kat Digital Corp.
3F-5, No.66, SanChong Rd. NanGang Taipei 115, Taiwan
Phone: 886 2 6617 3168
Website: www.katdc.com