Android

Ambrosi Michele
Rigo Francesco
Marcantoni Francesco

mastergap@gmail.com

4 giugno 2009

Indice

Cos' android?

Sistema operativo open-source per SmartPhones e dispositivi mobili ideato e sviluppato da Google e finanziato da Open Handset Alliance, una cooperazione tra diverse aziende del settore (hardware, software, compagnie telefoniche e compagnie di commercializzazione) tra le quali aziende molto importanti come Intel, NVidia, ARM holdings, Motorola, Sony-Ericsson, Toshiba, E-Bay, HTC, LG, Samsung Electronics.

Dispositivi

- Attualmente i dispositivi disponibili in commercio con Android installato al momento della distribuzione sono 3:
  - T-Mobile G1 (htc Dream)
  - T-Mobile G2 (htc Magic)
  - Samsung i7500



Dispositivi

- Diverse importanti compagnie internazionali hanno avviato progetti per la realizzazione di dispositivi di diverso tipo;
- Dell,HP $\longrightarrow$ netbook
- Acer, HTC $\longrightarrow$ smartphones
- Lenovo $\longrightarrow$ pda

Dispositivi

- Diverse importanti compagnie internazionali hanno avviato progetti per la realizzazione di dispositivi di diverso tipo;
- Dell,HP $\longrightarrow$ netbook
- Acer, HTC $\longrightarrow$ smartphones
- Lenovo $\longrightarrow$ pda

Android Market Developper Distribution on Agreement

- Documento che definisce i termini per la pubblicazione di applicazioni sull'Android Market;
- Consultabile all'indirizzo: http://www.android.com/us/developer-distribution-agreement.html;
- I punti di maggiore interesse sono quelli riguardanti guadagni/pagamenti e i diritti che Google acquisisce sulle applicazioni pubblicate;

Android Market Developper Distribution on Agreement

- 3.2 The price you set for Products will determine the amount of payment you will receive. A Transaction Fee, as defined below, will be charged on the sales price and apportioned to the Payment Processor and, if one exists, the Authorized Carrier. The remainder (sales price less Transaction Fee) will be remitted to you. The Transaction Fee is set forth at http://market.android.com/support/bin/answer.py?answer=112622 and may be revised by Google from time to time. Developer is responsible for determining if a Product is taxable and the applicable tax rate for the Payment Processor to collect for each taxing jurisdiction where Products are sold. Developer is responsible for remitting taxes to the appropriate taxing authority.
- For applications that you choose to sell in Android Market, the transaction fee is equivalent to 30% of the application price. For example, if you sell your application at a price of $10.00, the fee will be $3.00, and you will receive $7.00 in payment.

Android Market Developper Distribution on Agreement

- 3.3 You may also choose to distribute Products for free. If the Product is free, you will not be charged a Transaction Fee. You may not collect future charges from users for copies of the Products that those users were initially allowed to download for free. This is not intended to prevent distribution of free trial versions of the Product with an upsell option to obtain the full version of the Product: Such free trials for Products are encouraged. However, if you want to collect fees after the free trial expires, you must collect all fees for the full version of the Product through the Payment Processor on the Market. In this Agreement, free means there are no charges or fees of any kind for use of the Product. All fees received by Developers for Products distributed via the Market must be processed by the Markets Payment Processor.
- 3.6 Reinstalls. Users are allowed unlimited reinstalls of each application distributed via the Market.

Android Market Developper Distribution on Agreement

- 4.1 Except for the license rights granted by you in Section 5 below, Google agrees that it obtains no right, title or interest from you (or your licensors) under this Agreement in or to any of Products, including any intellectual property rights which subsist in those applications.
- 5.1 You grant to Google a nonexclusive, worldwide, and royalty-free license to: copy, perform, display, and use the Products for administrative and demonstration purposes in connection with the operation and marketing of the Market and to use the Products to make improvements to the Android platform.

Obiettivi Android

Due obiettivi principali:

- fornire agli sviluppatori di applicazioni dei tool di sviluppo potenti, semplici e completi
- rendere tutto il pi aperto ed esetensibile possibile, in modo da poter combinare tutte le parti gi esistenti, per svilupparne altre nuove, ottenendo possibilit di evoluzione incredibilmente ampie.

Obiettivi Android

Due obiettivi principali:

- fornire agli sviluppatori di applicazioni dei tool di sviluppo potenti, semplici e completi
- rendere tutto il pi aperto ed esetensibile possibile, in modo da poter combinare tutte le parti gi esistenti, per svilupparne altre nuove, ottenendo possibilit di evoluzione incredibilmente ampie.

Obiettivi Android

Due obiettivi principali:

- fornire agli sviluppatori di applicazioni dei tool di sviluppo potenti, semplici e completi
- rendere tutto il pi aperto ed esetensibile possibile, in modo da poter combinare tutte le parti gi esistenti, per svilupparne altre nuove, ottenendo possibilit di evoluzione incredibilmente ampie.

Kernel 2.6

- Si tratta di un kernel utilizzato principalmente da sistemi operativi Unix;
- La prima versione è stata rilasciata il 17 Dicembre 2003;
- L'ultima versione disponibile è la 2.6.29 rilasciata il 23 Marzo 2009;

Kernel 2.6

- Si tratta di un kernel utilizzato principalmente da sistemi operativi Unix;
- La prima versione è stata rilasciata il 17 Dicembre 2003;
- L'ultima versione disponibile è la 2.6.29 rilasciata il 23 Marzo 2009;

Kernel 2.6

- Si tratta di un kernel utilizzato principalmente da sistemi operativi Unix;
- La prima versione è stata rilasciata il 17 Dicembre 2003;
- L'ultima versione disponibile è la 2.6.29 rilasciata il 23 Marzo 2009;

Kernel 2.6 scheduler 1/2

- Utilizza un algoritmo a tempo costante, $O(1)$;
- Ad ogni CPU vengono assegnate 140 runqueues FIFO, una per ogni possibile grado di priorita dette *active queues* ;
- Le prime 100 runqueues sono riservate a task real-time mentre le rimanenti 40 sono riservate ai processi utente;
- Quando un task guadagna o perde priorità viene oppurtunamente assegnato ad una delle runqueues del processore su cui è andato in esecuzione l'ultima volta;
- Con una bitmap si indica quali runqueues sono vuote e quali no, viene quindi mandato in esecuzione il primo task della prima runqueue non vuota;
- Quando un task termina il proprio timeslice vine spostato in una delle 140 code parallele per processore dette *expired queues*;

Kernel 2.6 scheduler 1/2

- Utilizza un algoritmo a tempo costante, $O(1)$;
- Ad ogni CPU vengono assegnate 140 runqueues FIFO, una per ogni possibile grado di priorita dette *active queues* ;
- Le prime 100 runqueues sono riservate a task real-time mentre le rimanenti 40 sono riservate ai processi utente;
- Quando un task guadagna o perde priorità viene oppurtunamente assegnato ad una delle runqueues del processore su cui è andato in esecuzione l'ultima volta;
- Con una bitmap si indica quali runqueues sono vuote e quali no, viene quindi mandato in esecuzione il primo task della prima runqueue non vuota;
- Quando un task termina il proprio timeslice vine spostato in una delle 140 code parallele per processore dette *expired queues*;

Kernel 2.6 scheduler 1/2

- Utilizza un algoritmo a tempo costante, $O(1)$;
- Ad ogni CPU vengono assegnate 140 runqueues FIFO, una per ogni possibile grado di priorita dette *active queues* ;
- Le prime 100 runqueues sono riservate a task real-time mentre le rimanenti 40 sono riservate ai processi utente;
- Quando un task guadagna o perde priorità viene oppurtunamente assegnato ad una delle runqueues del processore su cui è andato in esecuzione l'ultima volta;
- Con una bitmap si indica quali runqueues sono vuote e quali no, viene quindi mandato in esecuzione il primo task della prima runqueue non vuota;
- Quando un task termina il proprio timeslice vine spostato in una delle 140 code parallele per processore dette *expired queues*;

Kernel 2.6 scheduler 1/2

- Utilizza un algoritmo a tempo costante, $O(1)$;
- Ad ogni CPU vengono assegnate 140 runqueues FIFO, una per ogni possibile grado di priorita dette *active queues* ;
- Le prime 100 runqueues sono riservate a task real-time mentre le rimanenti 40 sono riservate ai processi utente;
- Quando un task guadagna o perde priorità viene oppurtunamente assegnato ad una delle runqueues del processore su cui è andato in esecuzione l'ultima volta;
- Con una bitmap si indica quali runqueues sono vuote e quali no, viene quindi mandato in esecuzione il primo task della prima runqueue non vuota;
- Quando un task termina il proprio timeslice vine spostato in una delle 140 code parallele per processore dette *expired queues*;

Kernel 2.6 scheduler 1/2

- Utilizza un algoritmo a tempo costante, $O(1)$;
- Ad ogni CPU vengono assegnate 140 runqueues FIFO, una per ogni possibile grado di priorita dette *active queues* ;
- Le prime 100 runqueues sono riservate a task real-time mentre le rimanenti 40 sono riservate ai processi utente;
- Quando un task guadagna o perde priorità viene oppurtunamente assegnato ad una delle runqueues del processore su cui è andato in esecuzione l'ultima volta;
- Con una bitmap si indica quali runqueues sono vuote e quali no, viene quindi mandato in esecuzione il primo task della prima runqueue non vuota;
- Quando un task termina il proprio timeslice vine spostato in una delle 140 code parallele per processore dette *expired queues*;

Kernel 2.6 scheduler 1/2

- Utilizza un algoritmo a tempo costante, $O(1)$;
- Ad ogni CPU vengono assegnate 140 runqueues FIFO, una per ogni possibile grado di priorita dette *active queues* ;
- Le prime 100 runqueues sono riservate a task real-time mentre le rimanenti 40 sono riservate ai processi utente;
- Quando un task guadagna o perde priorità viene oppurtunamente assegnato ad una delle runqueues del processore su cui è andato in esecuzione l'ultima volta;
- Con una bitmap si indica quali runqueues sono vuote e quali no, viene quindi mandato in esecuzione il primo task della prima runqueue non vuota;
- Quando un task termina il proprio timeslice vine spostato in una delle 140 code parallele per processore dette *expired queues*;

Kernel 2.6 scheduler 2/2

- Quando una active queue resta vuota, una expired queue diventa runqueue con un semplice aggiornamento di un puntatore;
- Lo scheduler permette la prelazione: quando un processo ad alta priorità è pronto per essere mandato in esecuzione il processo in esecuzione se ha priorità minore viene prelazionato, rimesso nella propria run queue e si riesegue lo schedule;
- Le priorità dei task vengono mutate dinamicamente, i tasks I/O-bound sono privilegiati rispetto ai task CPU-bound;
- Si tende quindi ad assegnare un'alta priorità a processi con un alto grado di interattività con l'utente;
- Per sistemi multi-processore questo schema non prevede la migrazione di processi, viene quindi eseguito ogni 200ms un algoritmo di bilanciamento;

Kernel 2.6 scheduler 2/2

- Quando una active queue resta vuota, una expired queue diventa runqueue con un semplice aggiornamento di un puntatore;
- Lo scheduler permette la prelazione: quando un processo ad alta priorità è pronto per essere mandato in esecuzione il processo in esecuzione se ha priorità minore viene prelazionato, rimesso nella propria run queue e si riesegue lo schedule;
- Le priorità dei task vengono mutate dinamicamente, i tasks I/O-bound sono privilegiati rispetto ai task CPU-bound;
- Si tende quindi ad assegnare un'alta priorità a processi con un alto grado di interattività con l'utente;
- Per sistemi multi-processore questo schema non prevede la migrazione di processi, viene quindi eseguito ogni 200ms un algoritmo di bilanciamento;

Kernel 2.6 scheduler 2/2

- Quando una active queue resta vuota, una expired queue diventa runqueue con un semplice aggiornamento di un puntatore;
- Lo scheduler permette la prelazione: quando un processo ad alta priorità è pronto per essere mandato in esecuzione il processo in esecuzione se ha priorità minore viene prelazionato, rimesso nella propria run queue e si riesegue lo schedule;
- Le priorità dei task vengono mutate dinamicamente, i tasks I/O-bound sono privilegiati rispetto ai task CPU-bound;
- Si tende quindi ad assegnare un'alta priorità a processi con un alto grado di interattività con l'utente;
- Per sistemi multi-processore questo schema non prevede la migrazione di processi, viene quindi eseguito ogni 200ms un algoritmo di bilanciamento;

Kernel 2.6 scheduler 2/2

- Quando una active queue resta vuota, una expired queue diventa runqueue con un semplice aggiornamento di un puntatore;
- Lo scheduler permette la prelazione: quando un processo ad alta priorità è pronto per essere mandato in esecuzione il processo in esecuzione se ha priorità minore viene prelazionato, rimesso nella propria run queue e si riesegue lo schedule;
- Le priorità dei task vengono mutate dinamicamente, i tasks I/O-bound sono privilegiati rispetto ai task CPU-bound;
- Si tende quindi ad assegnare un'alta priorità a processi con un alto grado di interattività con l'utente;
- Per sistemi multi-processore questo schema non prevede la migrazione di processi, viene quindi eseguito ogni 200ms un algoritmo di bilanciamento;

Kernel 2.6 scheduler 2/2

- Quando una active queue resta vuota, una expired queue diventa runqueue con un semplice aggiornamento di un puntatore;
- Lo scheduler permette la prelazione: quando un processo ad alta priorità è pronto per essere mandato in esecuzione il processo in esecuzione se ha priorità minore viene prelazionato, rimesso nella propria run queue e si riesegue lo schedule;
- Le priorità dei task vengono mutate dinamicamente, i tasks I/O-bound sono privilegiati rispetto ai task CPU-bound;
- Si tende quindi ad assegnare un'alta priorità a processi con un alto grado di interattività con l'utente;
- Per sistemi multi-processore questo schema non prevede la migrazione di processi, viene quindi eseguito ogni 200ms un algoritmo di bilanciamento;

Gestione della memoria 1/2

- Gestione della memoria basata sulla paginazione:
  - Un segmento di memoria consiste in una o più pagine;
  - Indirizzi utilizzati dai processi sono virtuali, vengono tradotti in indirizzi fisici sulla base di un mapping gestito dal kernel;
  - Indirizzi virtuali diversi possono essere tradotti nello stesso indirizzo fisico.

Gestione della memoria 2/2

- Rispetto al kernel 2.4 attualmente è supportato il meccanismo di *reverse mapping* che consiste nel mantenere per ogni indirizzo fisico una lista concatenata di puntatori alle Page Table Entries (PTEs) di ogni processo che attualmente sta mappando quella pagina;
- Questo sistema consente di aggiornare velocemente le page table dei processi che mappano pagine che sono state spostate nell'area di swap;
- È possibile inoltre, grazie all'opzione di configurazione *Highmem PTE*, salvare le page table entries nella high memory, lasciando a disposizione una porzione maggiore della low memory al kernel;

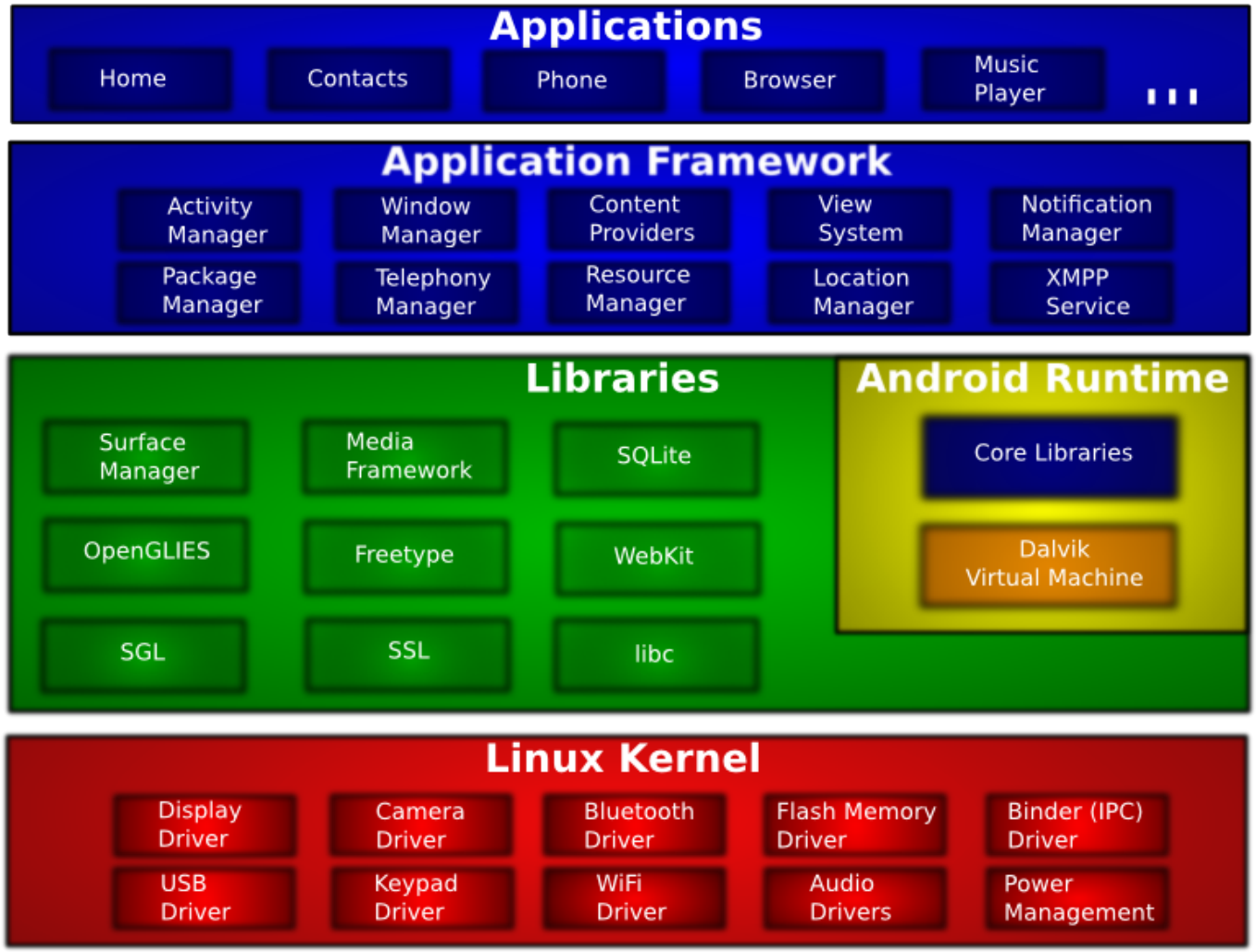
Gestione della memoria 2/2

- Rispetto al kernel 2.4 attualmente è supportato il meccanismo di *reverse mapping* che consiste nel mantenere per ogni indirizzo fisico una lista concatenata di puntatori alle Page Table Entries (PTEs) di ogni processo che attualmente sta mappando quella pagina;
- Questo sistema consente di aggiornare velocemente le page table dei processi che mappano pagine che sono state spostate nell'area di swap;
- È possibile inoltre, grazie all'opzione di configurazione *Highmem PTE*, salvare le page table entries nella high memory, lasciando a disposizione una porzione maggiore della low memory al kernel;

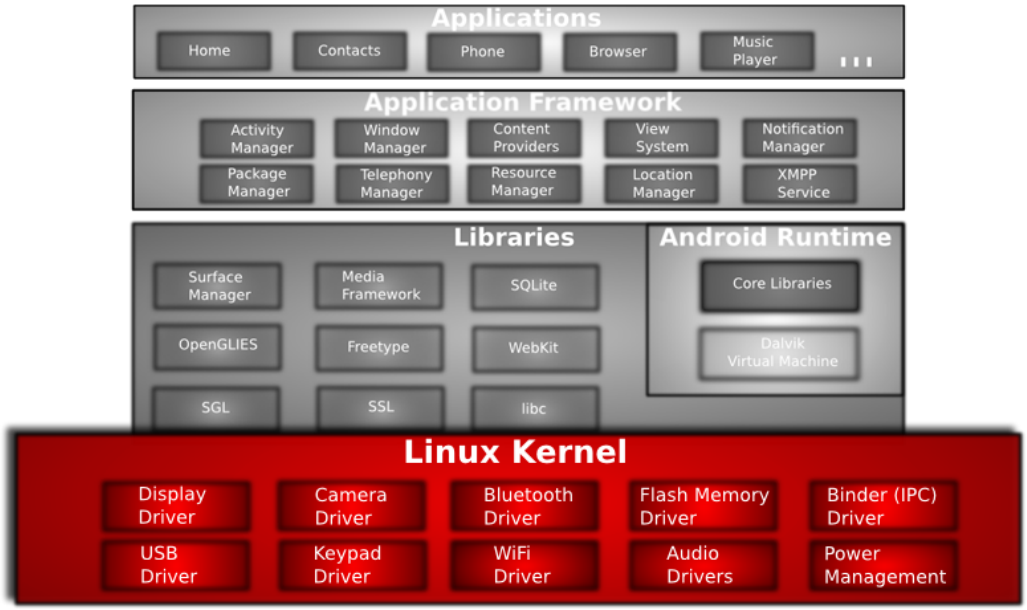
Gestione della memoria 2/2

- Rispetto al kernel 2.4 attualmente è supportato il meccanismo di *reverse mapping* che consiste nel mantenere per ogni indirizzo fisico una lista concatenata di puntatori alle Page Table Entries (PTEs) di ogni processo che attualmente sta mappando quella pagina;
- Questo sistema consente di aggiornare velocemente le page table dei processi che mappano pagine che sono state spostate nell'area di swap;
- È possibile inoltre, grazie all'opzione di configurazione *Highmem PTE*, salvare le page table entries nella high memory, lasciando a disposizione una porzione maggiore della low memory al kernel;

Schema architettura

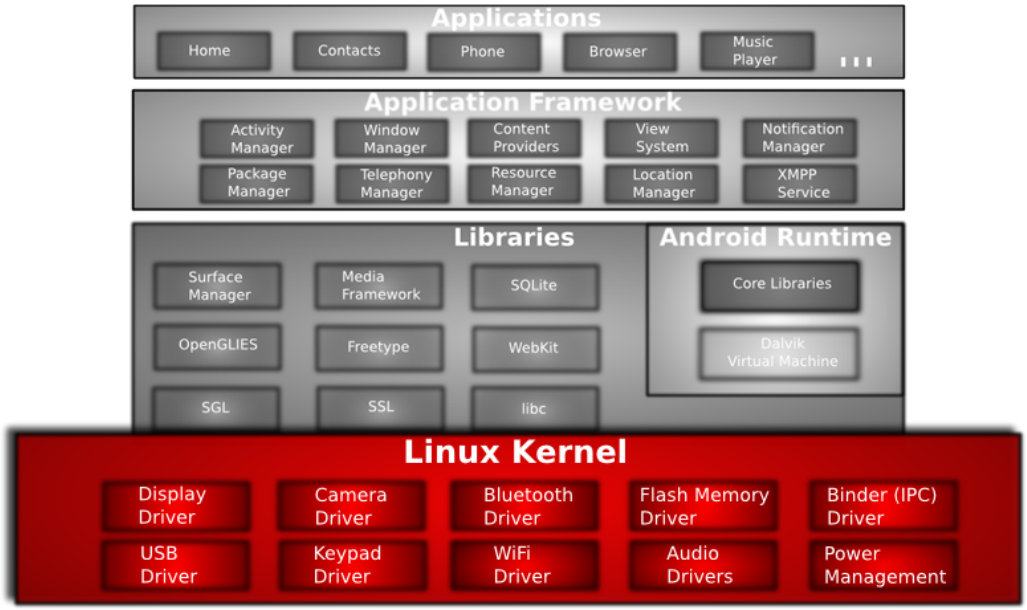


Livello 1: Linux Kernel



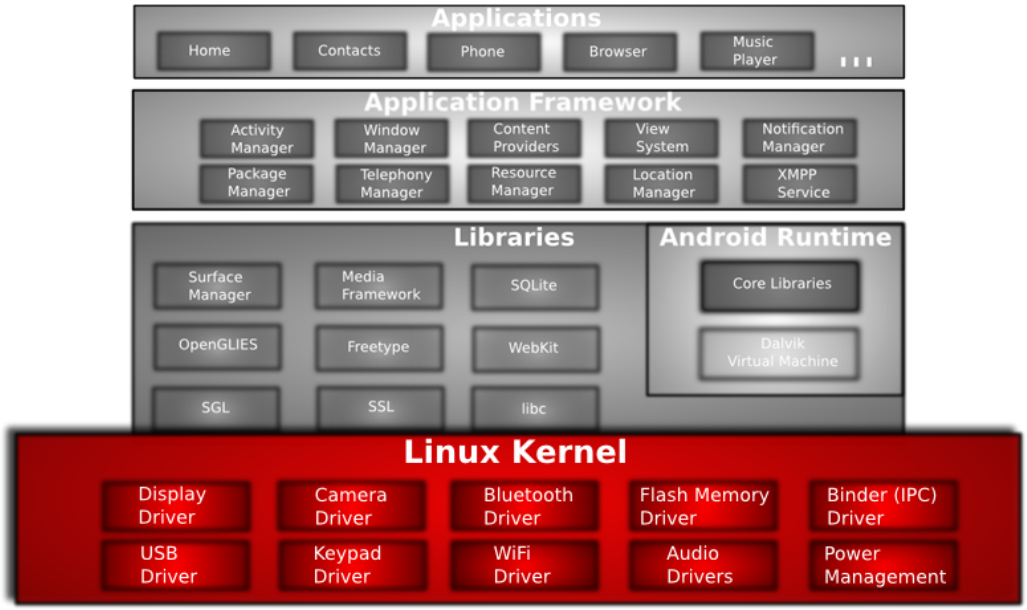
- Fornisce astrazione hardware.
- Se si vuole aggiungere supporto a nuovi dispositivi, basta aggiungere i rispettivi driver e moduli nel kernel.
- È stato usato un kernel linux perch fornisce gestione dei processi, gestione della memoria, modello di sicurezza, networking, etc. Inoltre robusto e in continua evoluzione.
- Attualmete viene utilizzato kernel 2.6

Livello 1: Linux Kernel



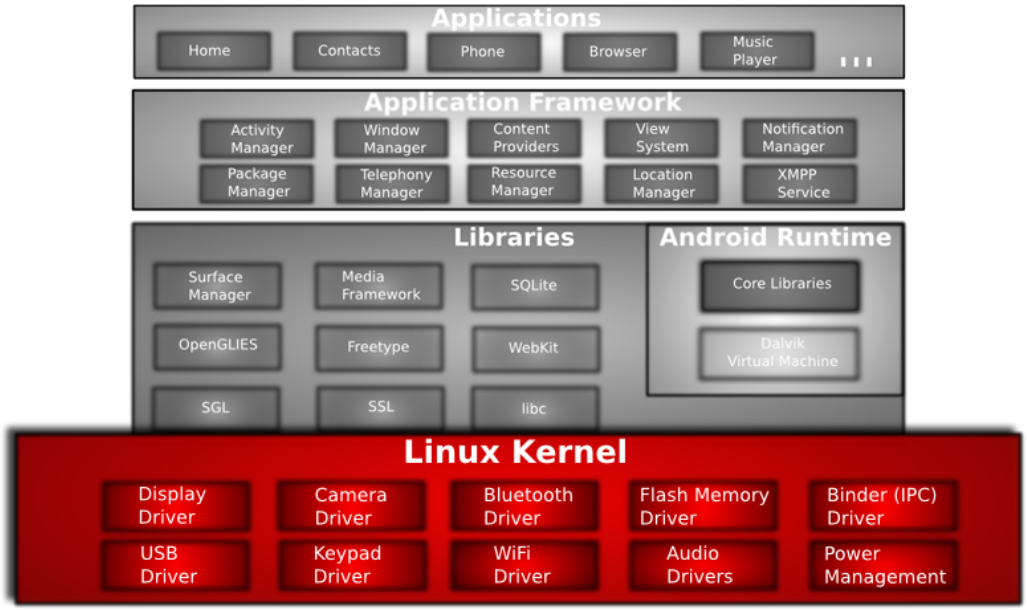
- Fornisce astrazione hardware.
- Se si vuole aggiungere supporto a nuovi dispositivi, basta aggiungere i rispettivi driver e moduli nel kernel.
- È stato usato un kernel linux perch fornisce gestione dei processi, gestione della memoria, modello di sicurezza, networking, etc. Inoltre robusto e in continua evoluzione.
- Attualmete viene utilizzato kernel 2.6

Livello 1: Linux Kernel



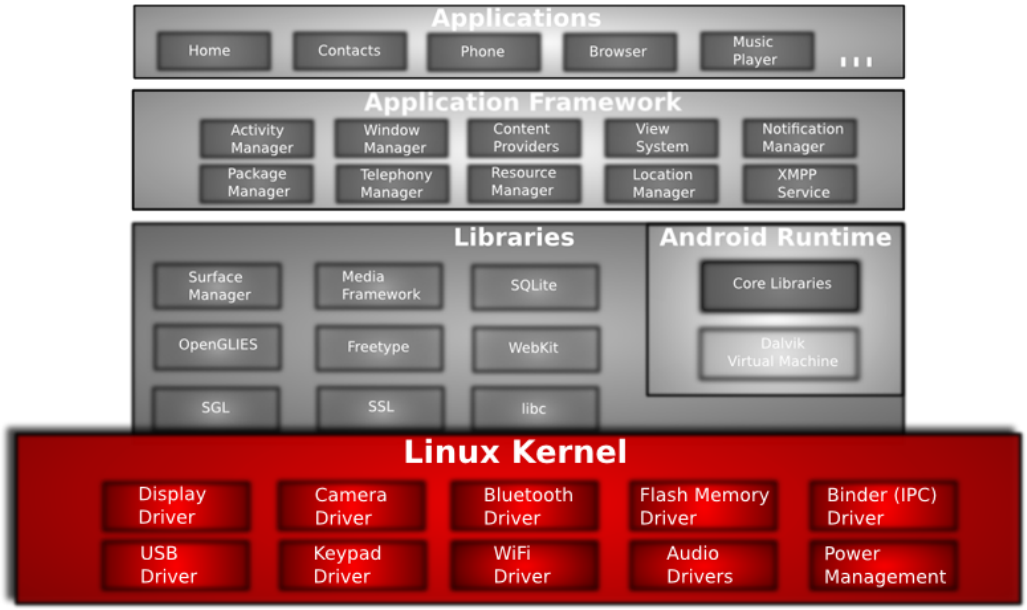
- Fornisce astrazione hardware.
- Se si vuole aggiungere supporto a nuovi dispositivi, basta aggiungere i rispettivi driver e moduli nel kernel.
- È stato usato un kernel linux perch fornisce gestione dei processi, gestione della memoria, modello di sicurezza, networking, etc. Inoltre robusto e in continua evoluzione.
- Attualmete viene utilizzato kernel 2.6

Livello 1: Linux Kernel



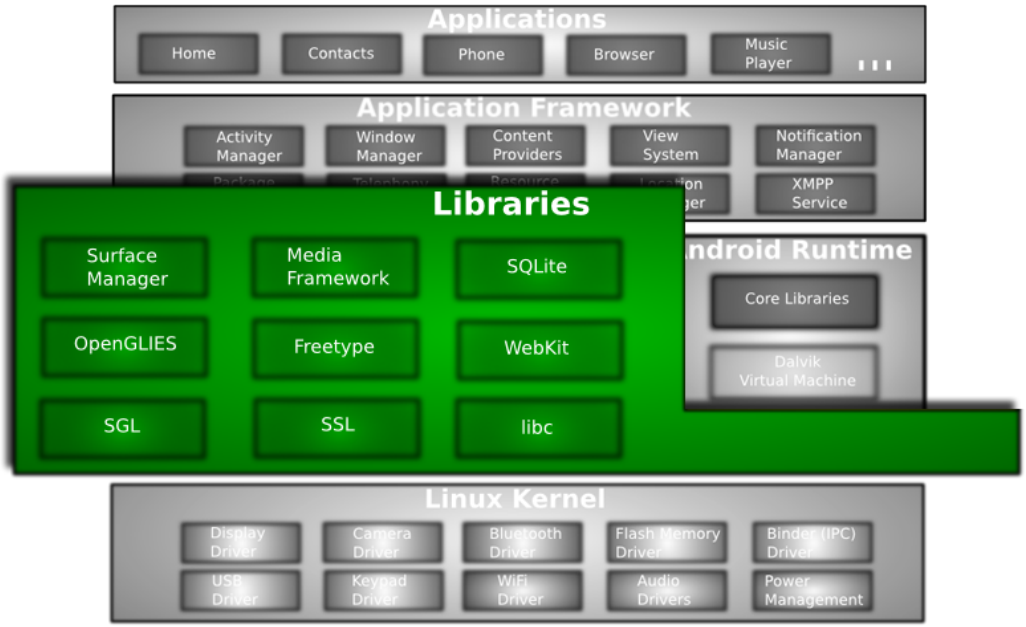
- Fornisce astrazione hardware.
- Se si vuole aggiungere supporto a nuovi dispositivi, basta aggiungere i rispettivi driver e moduli nel kernel.
- È stato usato un kernel linux perch fornisce gestione dei processi, gestione della memoria, modello di sicurezza, networking, etc. Inoltre robusto e in continua evoluzione.
- Attualmete viene utilizzato kernel 2.6

Livello 1: Linux Kernel



- Fornisce astrazione hardware.
- Se si vuole aggiungere supporto a nuovi dispositivi, basta aggiungere i rispettivi driver e moduli nel kernel.
- È stato usato un kernel linux perch fornisce gestione dei processi, gestione della memoria, modello di sicurezza, networking, etc. Inoltre robusto e in continua evoluzione.
- Attualmete viene utilizzato kernel 2.6

Livello 2-a: Libraries



Tutte le librerie di questo livello sono scritte in C e C++

Livello 2-a: Libraries -descrizione 1-

- **Surface Manager:** disegna differenti superfici grafiche sullo schermo. Gestisce finestre diverse appartenenti ad applicazioni diverse a loro volta appartenenti a processi diversi, tutte disegnate in tempi diversi, mette insieme i pixel e disegna tutto sullo schermo.
- **Librerie grafiche:**
  - **OpenGLIES:** libreria grafica 3D (supporto software) utilizzabile con chip di accelerazione 3D (supporto hardware)
  - **SGL:** libreria grafica 2D, praticamente tutte le applicazioni ne fanno largo utilizzo
  - Possibilit di combinare 2D e 3D nella stessa applicazione

Livello 2-a: Libraries -descrizione 1-

- **Surface Manager:** disegna differenti superfici grafiche sullo schermo. Gestisce finestre diverse appartenenti ad applicazioni diverse a loro volta appartenenti a processi diversi, tutte disegnate in tempi diversi, mette insieme i pixel e disegna tutto sullo schermo.
- **Librerie grafiche:**
  - **OpenGLIES:** libreria grafica 3D (supporto software) utilizzabile con chip di accelerazione 3D (supporto hardware)
  - **SGL:** libreria grafica 2D, praticamente tutte le applicazioni ne fanno largo utilizzo
  - Possibilit di combinare 2D e 3D nella stessa applicazione

Livello 2-a: Libraries -descrizione 2-

- **Media framework:** fornito da PacketVideo (uno dei membri della Open Handset Alliance). Contiene tutti i codecs per i formati multimediali maggiormente usati.
- **FreeType:** rendering dei fonts
- **SQLite:** implementazione database. Usato per salvare dati.
- **SSL:** gestisce comunicazioni sicure via socket
- **WebKit:** open source browser engine (lo stesso che sta alla base di Safari, browser di Apple). Ottimizzato per rendering delle pagine web su schermi piccoli.

Livello 2-a: Libraries -descrizione 2-

- **Media framework:** fornito da PacketVideo (uno dei membri della Open Handset Alliance). Contiene tutti i codecs per i formati multimediali maggiormente usati.
- **FreeType:** rendering dei fonts
- **SQLite:** implementazione database. Usato per salvare dati.
- **SSL:** gestisce comunicazioni sicure via socket
- **WebKit:** open source browser engine (lo stesso che sta alla base di Safari, browser di Apple). Ottimizzato per rendering delle pagine web su schermi piccoli.

Livello 2-a: Libraries -descrizione 2-

- **Media framework:** fornito da PacketVideo (uno dei membri della Open Handset Alliance). Contiene tutti i codecs per i formati multimediali maggiormente usati.
- **FreeType:** rendering dei fonts
- **SQLite:** implementazione database. Usato per salvare dati.
- **SSL:** gestisce comunicazioni sicure via socket
- **WebKit:** open source browser engine (lo stesso che sta alla base di Safari, browser di Apple). Ottimizzato per rendering delle pagine web su schermi piccoli.

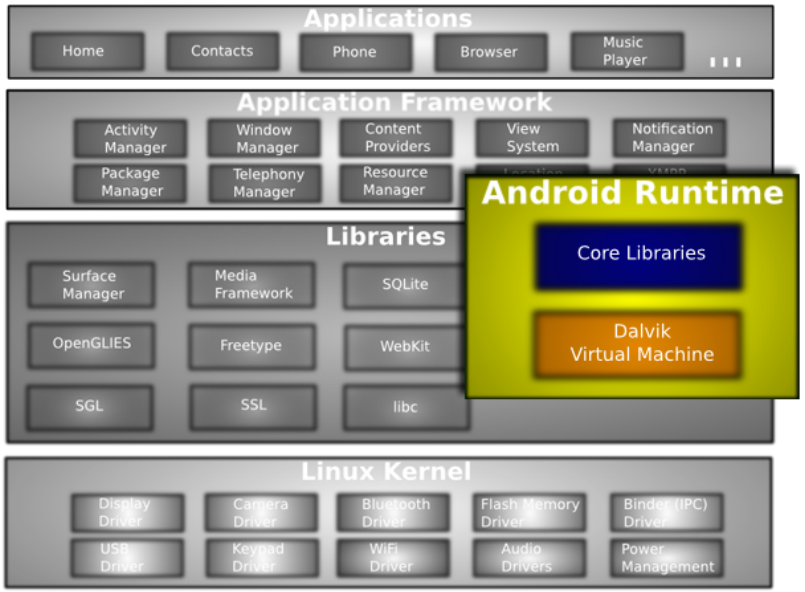
Livello 2-a: Libraries -descrizione 2-

- **Media framework:** fornito da PacketVideo (uno dei membri della Open Handset Alliance). Contiene tutti i codecs per i formati multimediali maggiormente usati.
- **FreeType:** rendering dei fonts
- **SQLite:** implementazione database. Usato per salvare dati.
- **SSL:** gestisce comunicazioni sicure via socket
- **WebKit:** open source browser engine (lo stesso che sta alla base di Safari, browser di Apple). Ottimizzato per rendering delle pagine web su schermi piccoli.

Livello 2-a: Libraries -descrizione 2-

- **Media framework:** fornito da PacketVideo (uno dei membri della Open Handset Alliance). Contiene tutti i codecs per i formati multimediali maggiormente usati.
- **FreeType:** rendering dei fonts
- **SQLite:** implementazione database. Usato per salvare dati.
- **SSL:** gestisce comunicazioni sicure via socket
- **WebKit:** open source browser engine (lo stesso che sta alla base di Safari, browser di Apple). Ottimizzato per rendering delle pagine web su schermi piccoli.

Livello 2-b: Android Runtime



Progettato appositamente per sistemi embedded (batterie ridotte, cpu ridotte, memoria ridotta)

Livello 2-b: Android Runtime -Dalvik Virtual Machine-

La componente principale la **Dalvik Virtual Machine:**

- esegue **DEX** files: bytecodes risultanti dalla conversione a buildtime di files .class in files .jar. Una volta convertiti i .jar in .DEX si ha un bytecode molto efficiente per piccoli processori:
  - usano memoria in modo molto efficiente
  - le strutture dati sono realizzate in modo da essere condivise tra processi (quando possibile)
  - usano interprete bytecode altamente ottimizzato in termini di utilizzo CPU
- C' la possibilit di eseguire pi istanze della Dalvik Virtual Machine allo stesso tempo (una per ogni processo server)

Livello 2-b: Android Runtime -Dalvik Virtual Machine-

La componente principale la **Dalvik Virtual Machine:**

- esegue **DEX** files: bytecodes risultanti dalla conversione a buildtime di files .class in files .jar. Una volta convertiti i .jar in .DEX si ha un bytecode molto efficiente per piccoli processori:
  - usano memoria in modo molto efficiente
  - le strutture dati sono realizzate in modo da essere condivise tra processi (quando possibile)
  - usano interprete bytecode altamente ottimizzato in termini di utilizzo CPU
- C' la possibilit di eseguire pi istanze della Dalvik Virtual Machine allo stesso tempo (una per ogni processo server)

Livello 2-b: Android Runtime -Dalvik Virtual Machine-

La componente principale la **Dalvik Virtual Machine:**

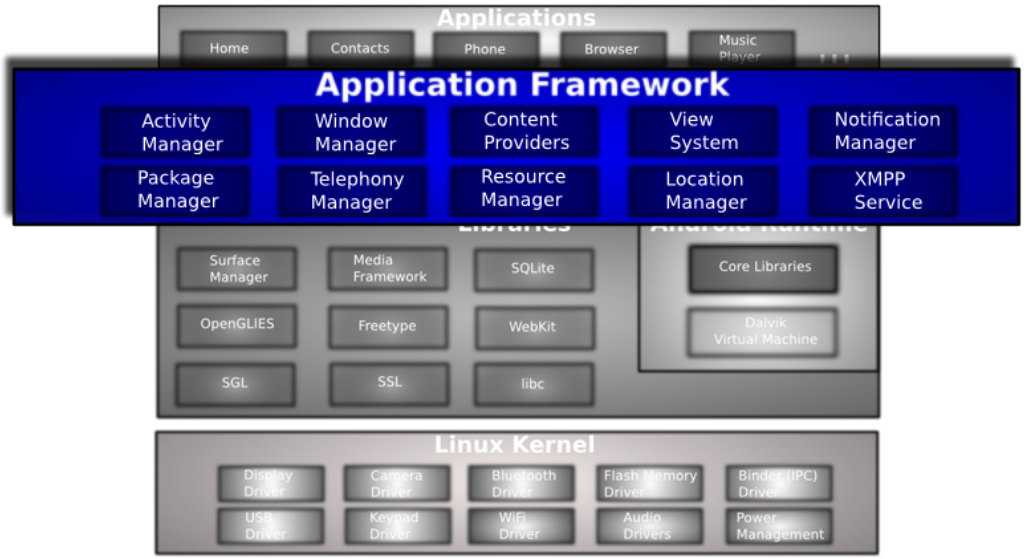
- esegue **DEX** files: bytecodes risultanti dalla conversione a buildtime di files .class in files .jar. Una volta convertiti i .jar in .DEX si ha un bytecode molto efficiente per piccoli processori:
  - usano memoria in modo molto efficiente
  - le strutture dati sono realizzate in modo da essere condivise tra processi (quando possibile)
  - usano interprete bytecode altamente ottimizzato in termini di utilizzo CPU
- C' la possibilit di eseguire pi istanze della Dalvik Virtual Machine allo stesso tempo (una per ogni processo server)

Livello 2-b: Android Runtime -Core Libraries-

L'altro componente sono le **Core Libraries:**

- sono scritte in linguaggio Java e contengono tutte le utilities necessarie alle applicazioni, come le Collection Classes, gestione I/O, etc.

Livello 3: Application Framework



Interamente scritto in linguaggio Java. È il toolkit utilizzato da ogni applicazione.

Livello 3: Application Framework -descrizione 1-

- **Activity Manager:** gestisce lifecycle di ogni applicazione. Mantiene command backstack in modo da integrare una navigazione fluida in applicazioni che girano su diversi processi
- **Package Manager:** mantiene traccia delle applicazioni installate nel dispositivo
- **Window Manager:** gestisce le finestre, si interfaccia con i servizi forniti ad un livello pi basso dal Surface Manager
- **Telephony Manager:** contiene APIs per sviluppare applicazioni riguardanti la telefonia
- **Content Providers:** permettono condivisione di dati tra le applicazioni (es. la rubrica pu essere accessibile a tutte le applicazioni)

Livello 3: Application Framework -descrizione 1-

- **Activity Manager:** gestisce lifecycle di ogni applicazione. Mantiene command backstack in modo da integrare una navigazione fluida in applicazioni che girano su diversi processi
- **Package Manager:** mantiene traccia delle applicazioni installate nel dispositivo
- **Window Manager:** gestisce le finestre, si interfaccia con i servizi forniti ad un livello pi basso dal Surface Manager
- **Telephony Manager:** contiene APIs per sviluppare applicazioni riguardanti la telefonia
- **Content Providers:** permettono condivisione di dati tra le applicazioni (es. la rubrica pu essere accessibile a tutte le applicazioni)

Livello 3: Application Framework -descrizione 1-

- **Activity Manager:** gestisce lifecycle di ogni applicazione. Mantiene command backstack in modo da integrare una navigazione fluida in applicazioni che girano su diversi processi
- **Package Manager:** mantiene traccia delle applicazioni installate nel dispositivo
- **Window Manager:** gestisce le finestre, si interfaccia con i servizi forniti ad un livello pi basso dal Surface Manager
- **Telephony Manager:** contiene APIs per sviluppare applicazioni riguardanti la telefonia
- **Content Providers:** permettono condivisione di dati tra le applicazioni (es. la rubrica pu essere accessibile a tutte le applicazioni)

Livello 3: Application Framework -descrizione 1-

- **Activity Manager:** gestisce lifecycle di ogni applicazione. Mantiene command backstack in modo da integrare una navigazione fluida in applicazioni che girano su diversi processi
- **Package Manager:** mantiene traccia delle applicazioni installate nel dispositivo
- **Window Manager:** gestisce le finestre, si interfaccia con i servizi forniti ad un livello pi basso dal Surface Manager
- **Telephony Manager:** contiene APIs per sviluppare applicazioni riguardanti la telefonia
- **Content Providers:** permettono condivisione di dati tra le applicazioni (es. la rubrica pu essere accessibile a tutte le applicazioni)

Livello 3: Application Framework -descrizione 1-

- **Activity Manager:** gestisce lifecycle di ogni applicazione. Mantiene command backstack in modo da integrare una navigazione fluida in applicazioni che girano su diversi processi
- **Package Manager:** mantiene traccia delle applicazioni installate nel dispositivo
- **Window Manager:** gestisce le finestre, si interfaccia con i servizi forniti ad un livello pi basso dal Surface Manager
- **Telephony Manager:** contiene APIs per sviluppare applicazioni riguardanti la telefonia
- **Content Providers:** permettono condivisione di dati tra le applicazioni (es. la rubrica pu essere accessibile a tutte le applicazioni)

Livello 3: Application Framework -descrizione 2-

- **Resource Manager:** utilizzato per salvare tutte le parti esterne di un'applicazione presenti nel codice (bitmaps, layout files, etc.)
- **View System:** contiene tutti gli elementi grafici come bottoni, checkboxes, etc. Contiene inoltre la gestione degli eventi e dei layout.
- **Location Manager, Notification Manager, XMPP service:** alcune APIs di supporto agli sviluppatori per creare applicazioni complesse e innovative.
  Ci soffermeremo pi tardi su queste APIs.

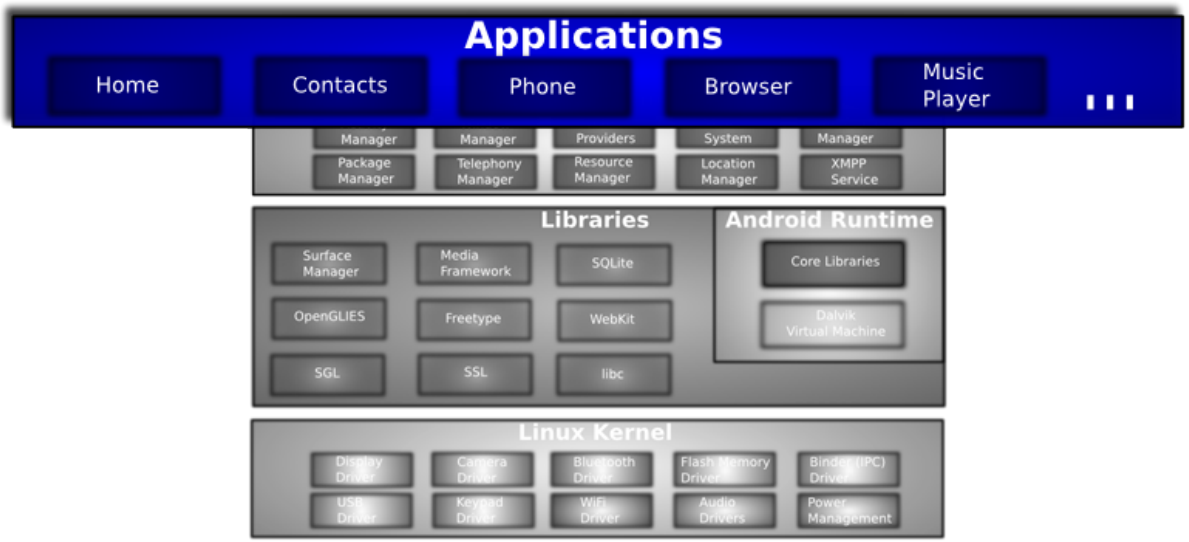
Livello 3: Application Framework -descrizione 2-

- **Resource Manager:** utilizzato per salvare tutte le parti esterne di un'applicazione presenti nel codice (bitmaps, layout files, etc.)
- **View System:** contiene tutti gli elementi grafici come bottoni, checkboxes, etc. Contiene inoltre la gestione degli eventi e dei layout.
- **Location Manager, Notification Manager, XMPP service:** alcune APIs di supporto agli sviluppatori per creare applicazioni complesse e innovative.
  Ci soffermeremo pi tardi su queste APIs.

Livello 3: Application Framework -descrizione 2-

- **Resource Manager:** utilizzato per salvare tutte le parti esterne di un'applicazione presenti nel codice (bitmaps, layout files, etc.)
- **View System:** contiene tutti gli elementi grafici come bottoni, checkboxes, etc. Contiene inoltre la gestione degli eventi e dei layout.
- **Location Manager, Notification Manager, XMPP service:** alcune APIs di supporto agli sviluppatori per creare applicazioni complesse e innovative.
  Ci soffermeremo pi tardi su queste APIs.

Livello 4: Application



Contiene tutte le applicazioni: quelle di base del telefono, quelle aggiuntive fornite da google, quelle sviluppate da terze parti. Tutte utilizzano lo stesso Application Framework.

Application Building Blocks

Ogni applicazione pu essere composta da pi parti fondamentali:

- **Activity:** componente UI tipicamente corrispondente a una schermata (es: applicazione controllo posta formata da 3 principali activities: lista delle mail, visualizzazione messaggio e composizione messaggio)
- **IntentReceiver:** risponde a notifiche o cambi di stato. Pu svegliare un processo. Si pu scrivere del codice che si sveglia e esegue quando avviene un determinato evento.
- **Service:** task invisibile (graficamente) che gira in background. Si pu interfacciare con delle activity con scambio di messaggi (es: ascoltare una playlist mentre si utilizzano altre applicazioni. Si pu fare un binding con un activity, ad esempio per cambiare canzone...)
- **ContentProvider:** permette alle applicazioni di condividere dati (es. la rubrica utilizza un ContentProvider in modo che ogni applicazione possa accedere ai contatti)

Application Building Blocks

Ogni applicazione pu essere composta da pi parti fondamentali:

- **Activity:** componente UI tipicamente corrispondente a una schermata (es: applicazione controllo posta formata da 3 principali activities: lista delle mail, visualizzazione messaggio e composizione messaggio)
- **IntentReceiver:** risponde a notifiche o cambi di stato. Pu svegliare un processo. Si pu scrivere del codice che si sveglia e esegue quando avviene un determinato evento.
- **Service:** task invisibile (graficamente) che gira in background. Si pu interfacciare con delle activity con scambio di messaggi (es: ascoltare una playlist mentre si utilizzano altre applicazioni. Si pu fare un binding con un activity, ad esempio per cambiare canzone...)
- **ContentProvider:** permette alle applicazioni di condividere dati (es. la rubrica utilizza un ContentProvider in modo che ogni applicazione possa accedere ai contatti)

Application Building Blocks

Ogni applicazione pu essere composta da pi parti fondamentali:

- **Activity:** componente UI tipicamente corrispondente a una schermata (es: applicazione controllo posta formata da 3 principali activities: lista delle mail, visualizzazione messaggio e composizione messaggio)
- **IntentReceiver:** risponde a notifiche o cambi di stato. Pu svegliare un processo. Si pu scrivere del codice che si sveglia e esegue quando avviene un determinato evento.
- **Service:** task invisibile (graficamente) che gira in background. Si pu interfacciare con delle activity con scambio di messaggi (es: ascoltare una playlist mentre si utilizzano altre applicazioni. Si pu fare un binding con un activity, ad esempio per cambiare canzone...)
- **ContentProvider:** permette alle applicazioni di condividere dati (es. la rubrica utilizza un ContentProvider in modo che ogni applicazione possa accedere ai contatti)

Application Building Blocks

Ogni applicazione pu essere composta da pi parti fondamentali:

- **Activity:** componente UI tipicamente corrispondente a una schermata (es: applicazione controllo posta formata da 3 principali activities: lista delle mail, visualizzazione messaggio e composizione messaggio)
- **IntentReceiver:** risponde a notifiche o cambi di stato. Pu svegliare un processo. Si pu scrivere del codice che si sveglia e esegue quando avviene un determinato evento.
- **Service:** task invisibile (graficamente) che gira in background. Si pu interfacciare con delle activity con scambio di messaggi (es: ascoltare una playlist mentre si utilizzano altre applicazioni. Si pu fare un binding con un activity, ad esempio per cambiare canzone...)
- **ContentProvider:** permette alle applicazioni di condividere dati (es. la rubrica utilizza un ContentProvider in modo che ogni applicazione possa accedere ai contatti)

Application Building Blocks

Ogni applicazione pu essere composta da pi parti fondamentali:

- **Activity:** componente UI tipicamente corrispondente a una schermata (es: applicazione controllo posta formata da 3 principali activities: lista delle mail, visualizzazione messaggio e composizione messaggio)
- **IntentReceiver:** risponde a notifiche o cambi di stato. Pu svegliare un processo. Si pu scrivere del codice che si sveglia e esegue quando avviene un determinato evento.
- **Service:** task invisibile (graficamente) che gira in background. Si pu interfacciare con delle activity con scambio di messaggi (es: ascoltare una playlist mentre si utilizzano altre applicazioni. Si pu fare un binding con un activity, ad esempio per cambiare canzone...)
- **ContentProvider:** permette alle applicazioni di condividere dati (es. la rubrica utilizza un ContentProvider in modo che ogni applicazione possa accedere ai contatti)

Rimpiazzare e riutilizzare componenti

- Il sistema seleziona il componente migliore in grado di soddisfare la richiesta

- ad esempio Photo Gallery.
- È possibile cambiare il componente migliore in qualsiasi momento

- ad esempio sostituire Photo Gallery con Picasa

Application Lifecycle

Ogni applicazione esegue in un proprio processo. Questo garantisce i seguenti vantaggi:

- sicurezza
- memoria protetta
- garantisce che applicazioni che fanno uso intensivo della cpu non blocchino altre applicazioni critiche (es. accesso alle funzioni telefoniche)

I processi possono essere avviati/stoppati, se necessario, per eseguire dei componenti di un'applicazione

I processi possono essere uccisi per richiesta di risorse

Application Lifecycle

Ogni applicazione esegue in un proprio processo. Questo garantisce i seguenti vantaggi:

- sicurezza
- memoria protetta
- garantisce che applicazioni che fanno uso intensivo della cpu non blocchino altre applicazioni critiche (es. accesso alle funzioni telefoniche)

I processi possono essere avviati/stoppati, se necessario, per eseguire dei componenti di un'applicazione

I processi possono essere uccisi per richiesta di risorse

Application Lifecycle

Ogni applicazione esegue in un proprio processo. Questo garantisce i seguenti vantaggi:

- sicurezza
- memoria protetta
- garantisce che applicazioni che fanno uso intensivo della cpu non blocchino altre applicazioni critiche (es. accesso alle funzioni telefoniche)

I processi possono essere avviati/stoppati, se necessario, per eseguire dei componenti di un'applicazione

I processi possono essere uccisi per richiesta di risorse

Application Lifecycle

Ogni applicazione esegue in un proprio processo. Questo garantisce i seguenti vantaggi:

- sicurezza
- memoria protetta
- garantisce che applicazioni che fanno uso intensivo della cpu non blocchino altre applicazioni critiche (es. accesso alle funzioni telefoniche)

I processi possono essere avviati/stoppati, se necessario, per eseguire dei componenti di un'applicazione

I processi possono essere uccisi per richiesta di risorse

Application Lifecycle

Ogni applicazione esegue in un proprio processo. Questo garantisce i seguenti vantaggi:

- sicurezza
- memoria protetta
- garantisce che applicazioni che fanno uso intensivo della cpu non blocchino altre applicazioni critiche (es. accesso alle funzioni telefoniche)

I processi possono essere avviati/stoppati, se necessario, per eseguire dei componenti di un'applicazione
I processi possono essere uccisi per richiesta di risorse

Esempio di navigazione tra applicazioni

Vedremo ora un esempio di come viene gestita la navigazione tra varie applicazioni da parte del sistema, con l'appoggio dell'Activity Manager, in modo del tutto trasparente all'utente.

Esempio:
Navigazione da Home (Home application) e inbox messages (Mail application), selezione e visualizzazione di un messaggio, selezione e apertura di un link contenuto nel messaggio (Browser), click su elemento per visualizzazione di una mappa (Maps application)

Esempio di navigazione tra applicazioni 1

Processo Home in esecuzione con relativa Activity.

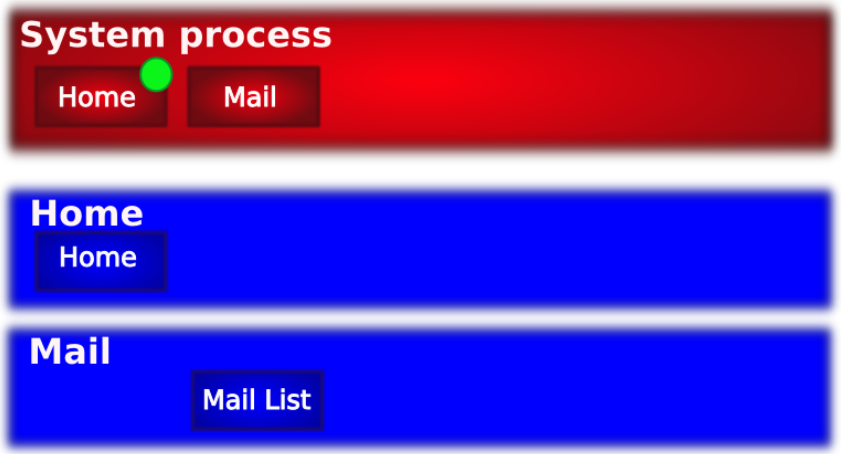
Abbiamo la schermata su Home dalla quale selezioniamo l'applicazione per la gestione della mail.

Esempio di navigazione tra applicazioni 2

Quando si passa ad un'altra applicazione (o Activity) l'applicazione precedente salva il suo stato (utilizzo del backstack).

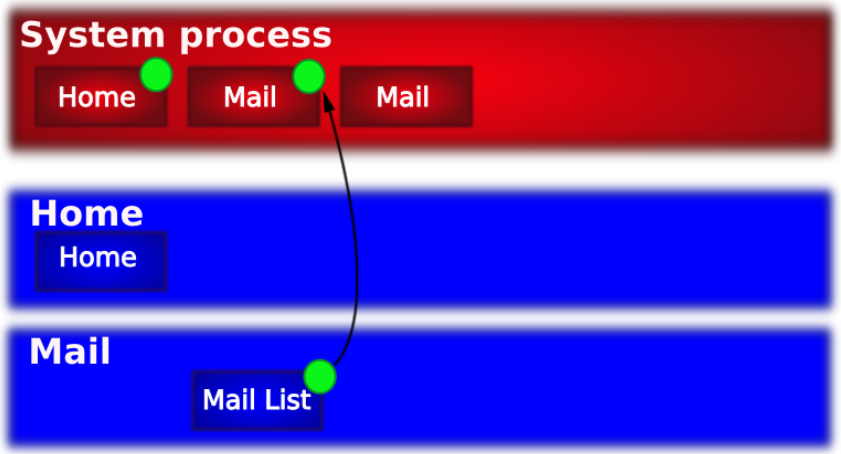
C' una richiesta per aprire Mail. Home salva il suo stato.

Esempio di navigazione tra applicazioni 3



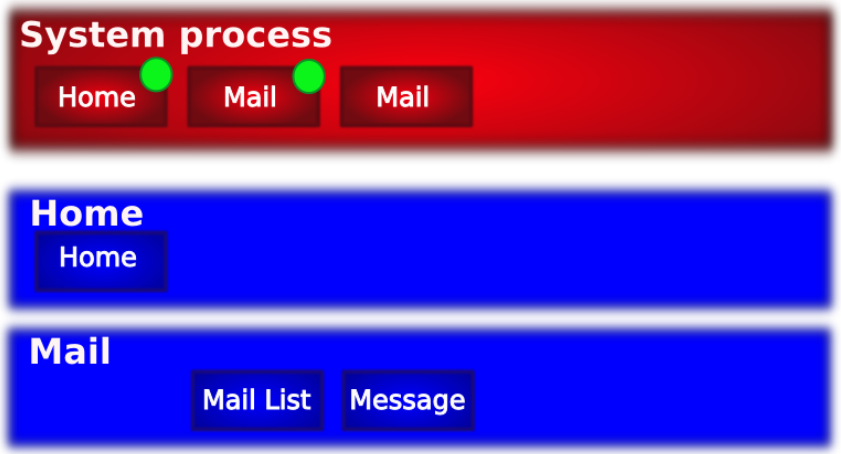
Viene creato un nuovo processo per Mail con la relativa Activity per visualizzare la lista di mail nella casella di posta in arrivo.

Esempio di navigazione tra applicazioni 4



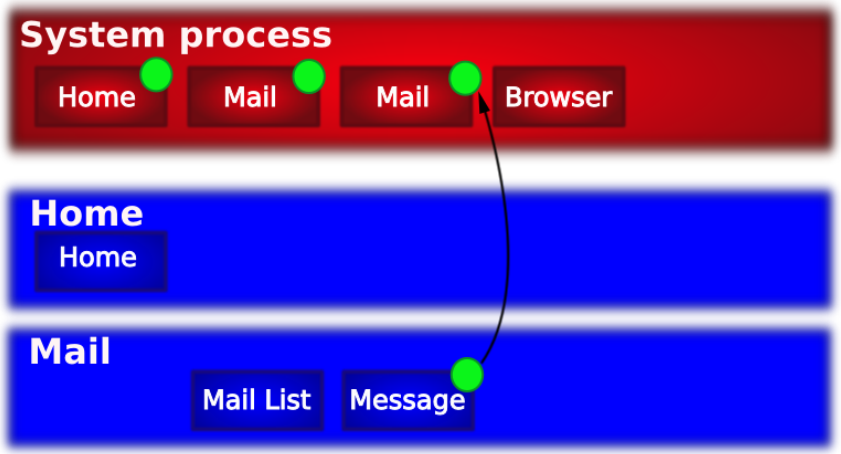
C' un'altra richiesta da parte del processo Mail, quindi il processo in esecuzione salva il suo stato.

Esempio di navigazione tra applicazioni 5



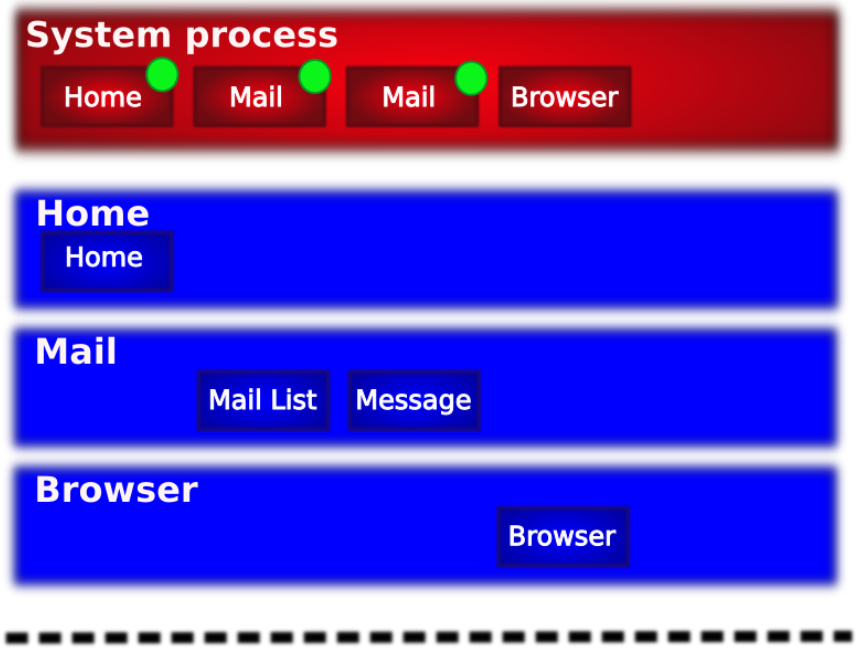
Il processo Mail gi in esecuzione quindi viene solo creata un'Activity al suo interno per visualizzare il contenuto di un messaggio selezionato.

Esempio di navigazione tra applicazioni 6



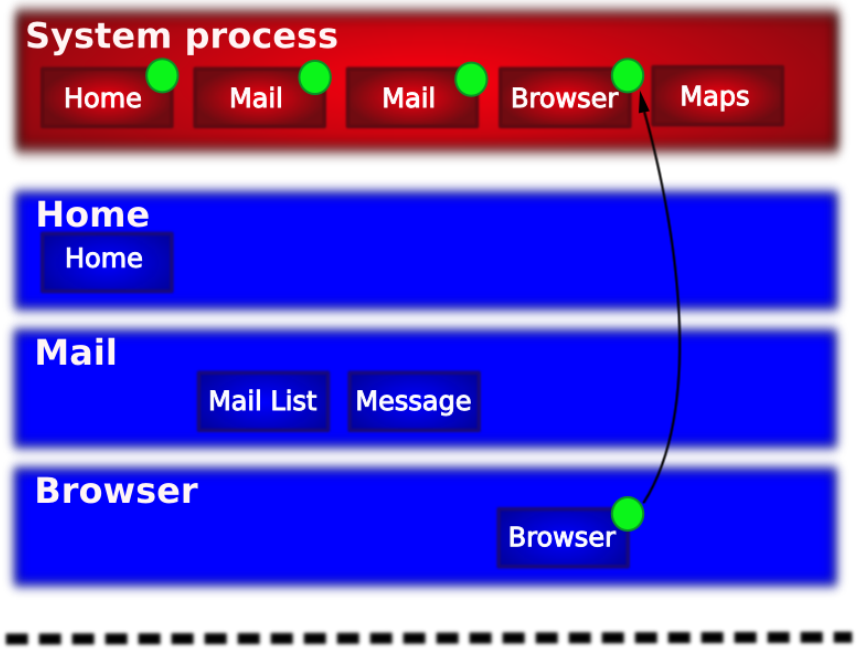
Selezionando un link nel messaggio viene creata una richiesta per aprire il browser web. Quindi Mail salva il suo stato.

Esempio di navigazione tra applicazioni 7



Viene creato un nuovo processo per Browser con la relativa Activity per visualizzare l'interfaccia e la pagina web.

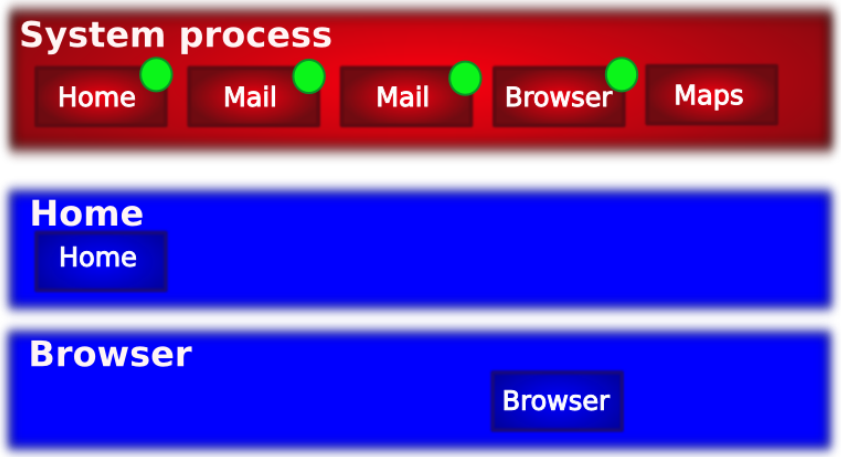
Esempio di navigazione tra applicazioni 8



Cliccando su un link ad una mappa abbiamo una richiesta per aprire una mappa.

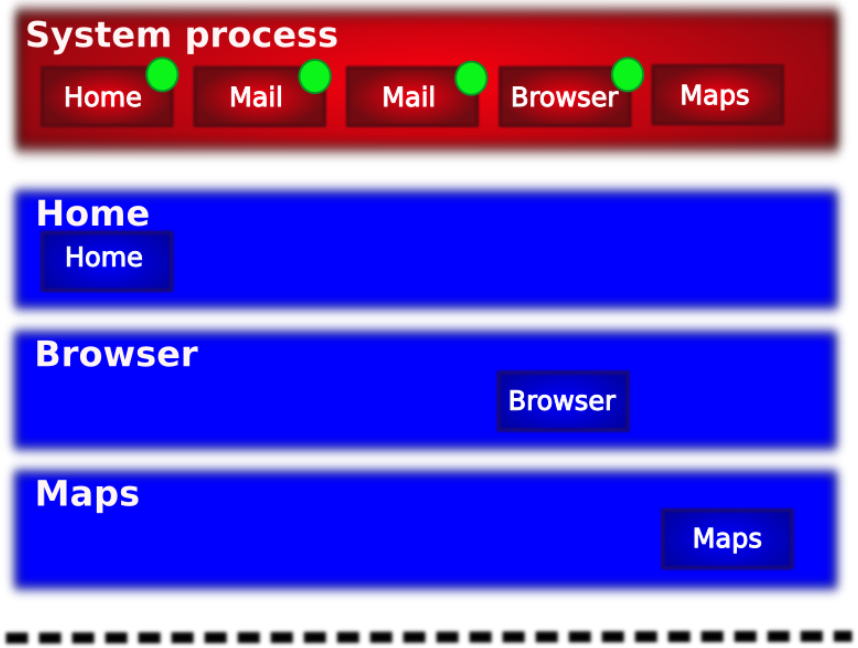
Browser salva il suo stato.

Esempio di navigazione tra applicazioni 9



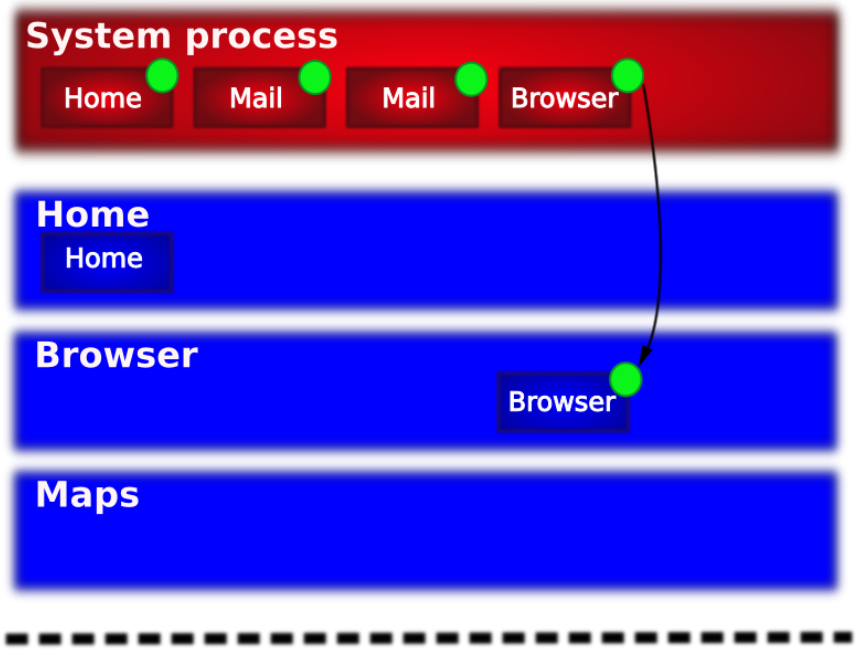
Non c' spazio per creare un altro processo in memoria. Quindi si deve ucciderne uno. Home non si pu uccidere, browser quello attualmente in esecuzione e visualizzato sul display, quindi si uccide Mail.

Esempio di navigazione tra applicazioni 10



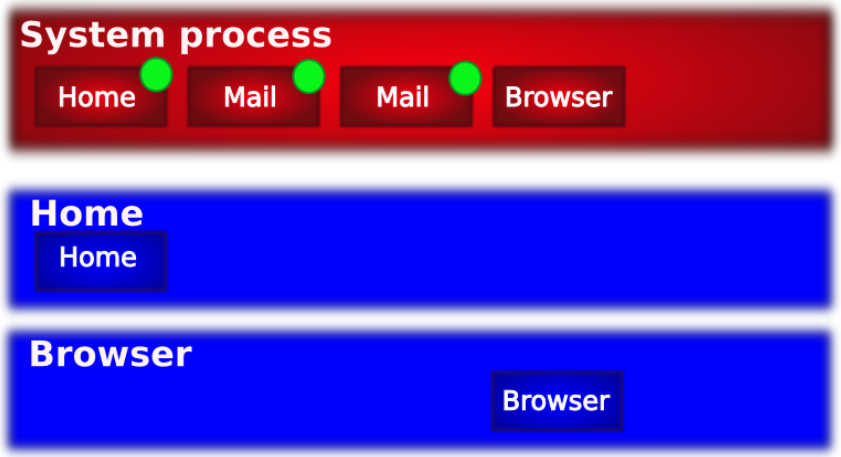
Viene creato un nuovo processo per Maps con la relativa Activity per vissualizzare la mappa.

Esempio di navigazione tra applicazioni 11



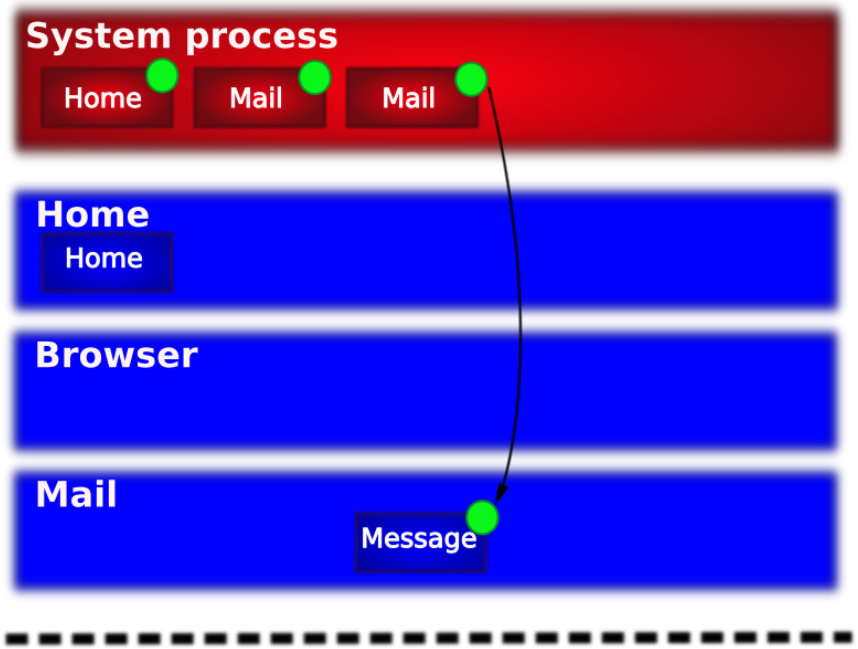
Ora si pu vedere come viene utilizzato il backstack per navigare all'indietro.
L'Activity del processo Maps viene chiusa e viene caricato lo stato per Browser.

Esempio di navigazione tra applicazioni 12



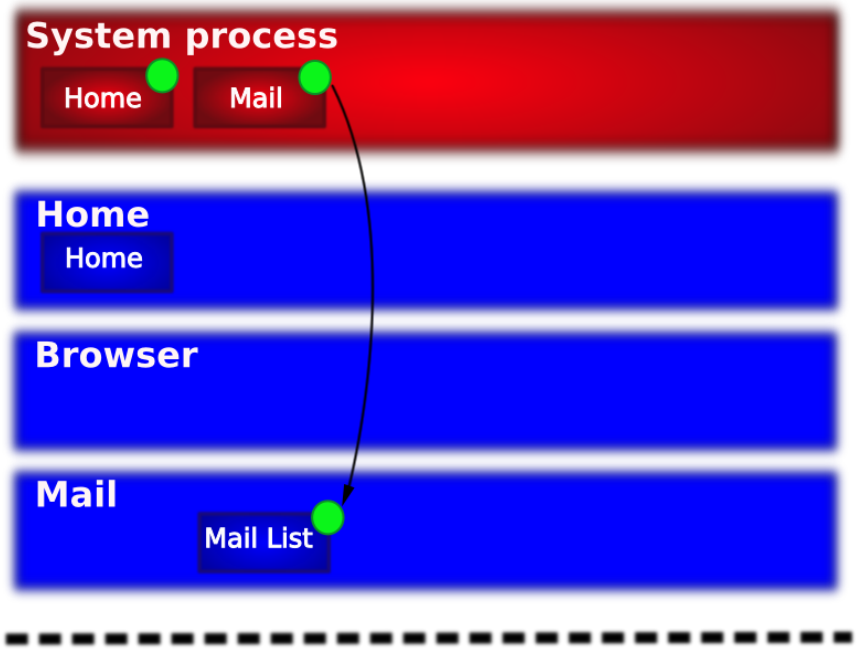
Il prcesso Mail non pi in esecuzione, e non c' spazio in memoria per crearlo. Si deve uccidere un processo, generalmente si uccide quello che compare pi lontano nel backstack. Viene ucciso Maps.

Esempio di navigazione tra applicazioni 13



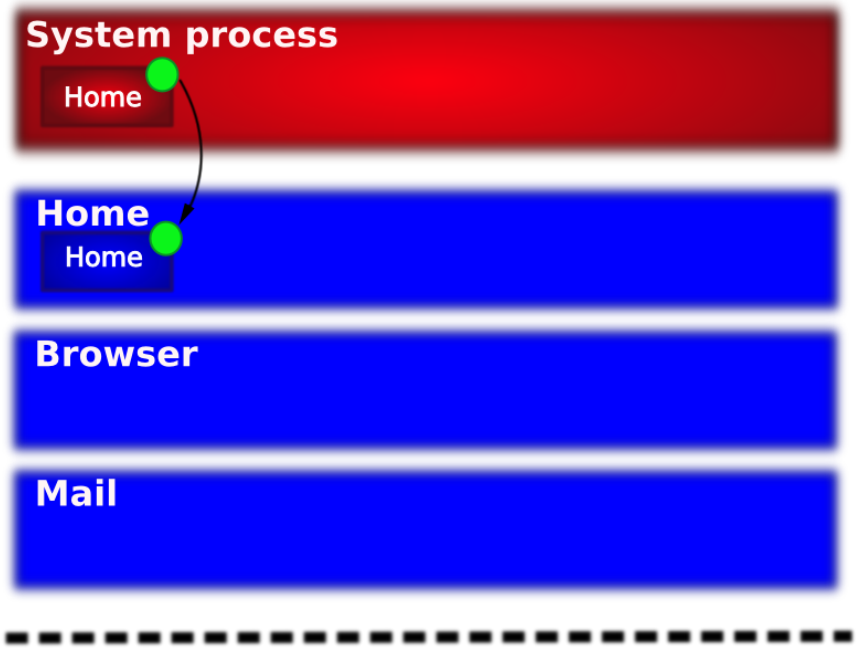
Viene chiusa l'Activity Browser, creata una nuova istanza del processo Mail, una nuova istanza dell'Activity Message e caricato lo stato salvato in precedenza.

Esempio di navigazione tra applicazioni 14



Viene chiusa l'Activity Message, creata una nuova istanza dell'Activity Mail List e caricato lo stato salvato in precedenza.

Esempio di navigazione tra applicazioni 15



Viene chiusa l'Activity Mail List e caricato lo stato salvato da Home.

Location Manager

Location Manager fornisce informazioni geografiche:

- le applicazioni possono registrarsi per la notifica della geo-posizione corrente
- possono essere registrati degli Intents che possono essere attivati basandosi sulla prossimit
- supporto principale per GPS, ma le informazioni possono essere reperite, nel caso di assenza di segnale gps, tramite ID di cella o tramite reti wifi aventi informazioni relative alla geolocalizzazione attraverso le quali possono cercare di approssimare la posizione (metodi alternativi=scarsa precisione)

Location Manager

Location Manager fornisce informazioni geografiche:

- le applicazioni possono registrarsi per la notifica della geo-posizione corrente
- possono essere registrati degli Intents che possono essere attivati basandosi sulla prossimit
- supporto principale per GPS, ma le informazioni possono essere reperite, nel caso di assenza di segnale gps, tramite ID di cella o tramite reti wifi aventi informazioni relative alla geolocalizzazione attraverso le quali possono cercare di approssimare la posizione (metodi alternativi=scarsa precisione)

Location Manager

Location Manager fornisce informazioni geografiche:

- le applicazioni possono registrarsi per la notifica della geo-posizione corrente
- possono essere registrati degli Intents che possono essere attivati basandosi sulla prossimit
- supporto principale per GPS, ma le informazioni possono essere reperite, nel caso di assenza di segnale gps, tramite ID di cella o tramite reti wifi aventi informazioni relative alla geolocalizzazione attraverso le quali possono cercare di approssimare la posizione (metodi alternativi=scarsa precisione)

Location Manager

Location Manager fornisce informazioni geografiche:

- le applicazioni possono registrarsi per la notifica della geo-posizione corrente
- possono essere registrati degli Intents che possono essere attivati basandosi sulla prossimit
- supporto principale per GPS, ma le informazioni possono essere reperite, nel caso di assenza di segnale gps, tramite ID di cella o tramite reti wifi aventi informazioni relative alla geolocalizzazione attraverso le quali possono cercare di approssimare la posizione (metodi alternativi=scarsa precisione)

XMPP Service

- permette ad ogni applicazione di mandare messaggi dati device-to-device ad altri utenti android
- messaggi dati sono Intents con una coppia (nome,valore)
- funziona con qualsiasi account Gmail
- supporta la possibilit di creare server per distribuire messaggi dati server-to-device

XMPP Service

- permette ad ogni applicazione di mandare messaggi dati device-to-device ad altri utenti android
- messaggi dati sono Intents con una coppia (nome,valore)
- funziona con qualsiasi account Gmail
- supporta la possibilit di creare server per distribuire messaggi dati server-to-device

XMPP Service

- permette ad ogni applicazione di mandare messaggi dati device-to-device ad altri utenti android
- messaggi dati sono Intents con una coppia (nome,valore)
- funziona con qualsiasi account Gmail
- supporta la possibilit di creare server per distribuire messaggi dati server-to-device

XMPP Service

- permette ad ogni applicazione di mandare messaggi dati device-to-device ad altri utenti android
- messaggi dati sono Intents con una coppia (nome,valore)
- funziona con qualsiasi account Gmail
- supporta la possibilit di creare server per distribuire messaggi dati server-to-device

Notification Manager

- permette ad ogni applicazione di aggiungere notifiche nella status bar
- la status bar pu essere utilizzata come per dare informazioni all'utente (es. ricevuto SMS, ricevuta mail, etc.)
- notifiche possono avere associata un'azione (Intent)

Notification Manager

- permette ad ogni applicazione di aggiungere notifiche nella status bar
- la status bar pu essere utilizzata come per dare informazioni all'utente (es. ricevuto SMS, ricevuta mail, etc.)
- notifiche possono avere associata un'azione (Intent)

Notification Manager

- permette ad ogni applicazione di aggiungere notifiche nella status bar
- la status bar pu essere utilizzata come per dare informazioni all'utente (es. ricevuto SMS, ricevuta mail, etc.)
- notifiche possono avere associata un'azione (Intent)

Views

- mette a disposizione diverse views (organizzazioni grafiche) come listbox, gallery view, widget buttons, etc.
- possono essere combinate tra loro
- supportano metodi di input multipli e dimensioni schermo multiple. Gli sviluppatori di applicazioni non si devono preoccupare di questi aspetti.
- permette sviluppo di applicazioni pi veloce
- facilita sviluppo GUI per applicazioni
- particolarmente interessanti Map View e Web View che permettono ripsettivamente di integrare in un applicazione una mappa o una pagina web con relativi meccanismi di navigazione del contenuto

Views

- mette a disposizione diverse views (organizzazioni grafiche) come listbox, gallery view, widget buttons, etc.
- possono essere combinate tra loro
- supportano metodi di input multipli e dimensioni schermo multiple. Gli sviluppatori di applicazioni non si devono preoccupare di questi aspetti.
- permette sviluppo di applicazioni pi veloce
- facilita sviluppo GUI per applicazioni
- particolarmente interessanti Map View e Web View che permettono ripsettivamente di integrare in un applicazione una mappa o una pagina web con relativi meccanismi di navigazione del contenuto

Views

- mette a disposizione diverse views (organizzazioni grafiche) come listbox, gallery view, widget buttons, etc.
- possono essere combinate tra loro
- supportano metodi di input multipli e dimensioni schermo multiple. Gli sviluppatori di applicazioni non si devono preoccupare di questi aspetti.
- permette sviluppo di applicazioni pi veloce
- facilita sviluppo GUI per applicazioni
- particolarmente interessanti Map View e Web View che permettono ripsettivamente di integrare in un applicazione una mappa o una pagina web con relativi meccanismi di navigazione del contenuto

Views

- mette a disposizione diverse views (organizzazioni grafiche) come listbox, gallery view, widget buttons, etc.
- possono essere combinate tra loro
- supportano metodi di input multipli e dimensioni schermo multiple. Gli sviluppatori di applicazioni non si devono preoccupare di questi aspetti.
- permette sviluppo di applicazioni pi veloce
- facilita sviluppo GUI per applicazioni
- particolarmente interessanti Map View e Web View che permettono ripsettivamente di integrare in un applicazione una mappa o una pagina web con relativi meccanismi di navigazione del contenuto

Views

- mette a disposizione diverse views (organizzazioni grafiche) come listbox, gallery view, widget buttons, etc.
- possono essere combinate tra loro
- supportano metodi di input multipli e dimensioni schermo multiple. Gli sviluppatori di applicazioni non si devono preoccupare di questi aspetti.
- permette sviluppo di applicazioni pi veloce
- facilita sviluppo GUI per applicazioni
- particolarmente interessanti Map View e Web View che permettono ripsettivamente di integrare in un applicazione una mappa o una pagina web con relativi meccanismi di navigazione del contenuto

Views

- mette a disposizione diverse views (organizzazioni grafiche) come listbox, gallery view, widget buttons, etc.
- possono essere combinate tra loro
- supportano metodi di input multipli e dimensioni schermo multiple. Gli sviluppatori di applicazioni non si devono preoccupare di questi aspetti.
- permette sviluppo di applicazioni pi veloce
- facilita sviluppo GUI per applicazioni
- particolarmente interessanti Map View e Web View che permettono ripsettivamente di integrare in un applicazione una mappa o una pagina web con relativi meccanismi di navigazione del contenuto