



Хакер - Android 1 глазами юзера Android 8. Какой была первая версия самой популярной ОС и как она менялась со временем
nopaywall



<https://t.me/nopaywall>

[Денис Погребной](#)

Содержание статьи

- [Android m3 и m5](#)
- [Краткая история версий](#)
- [Android 1.0–1.5](#)
- [Как менялся интерфейс Android](#)
- [Технические аспекты](#)
- [Парадокс новизны и безопасности](#)
- [Старый софт](#)
- [Opera](#)
- [SlideScreen](#)
- [Facebook](#)
- [ES File Explorer](#)
- [Заключение](#)

Десять лет назад программисты Google выложили в открытый доступ первый SDK Android и первый эмулятор, с помощью которого можно было оценить операционную систему. Тогда мало кто воспринял новую ОС серьезно. Тем интереснее посмотреть на первый Android сегодня. Мы откопали все «альфы» и «беты» Android десятилетней давности, тщательно их протестировали, расковыряли внутренности и теперь готовы рассказать о том, что нашли.

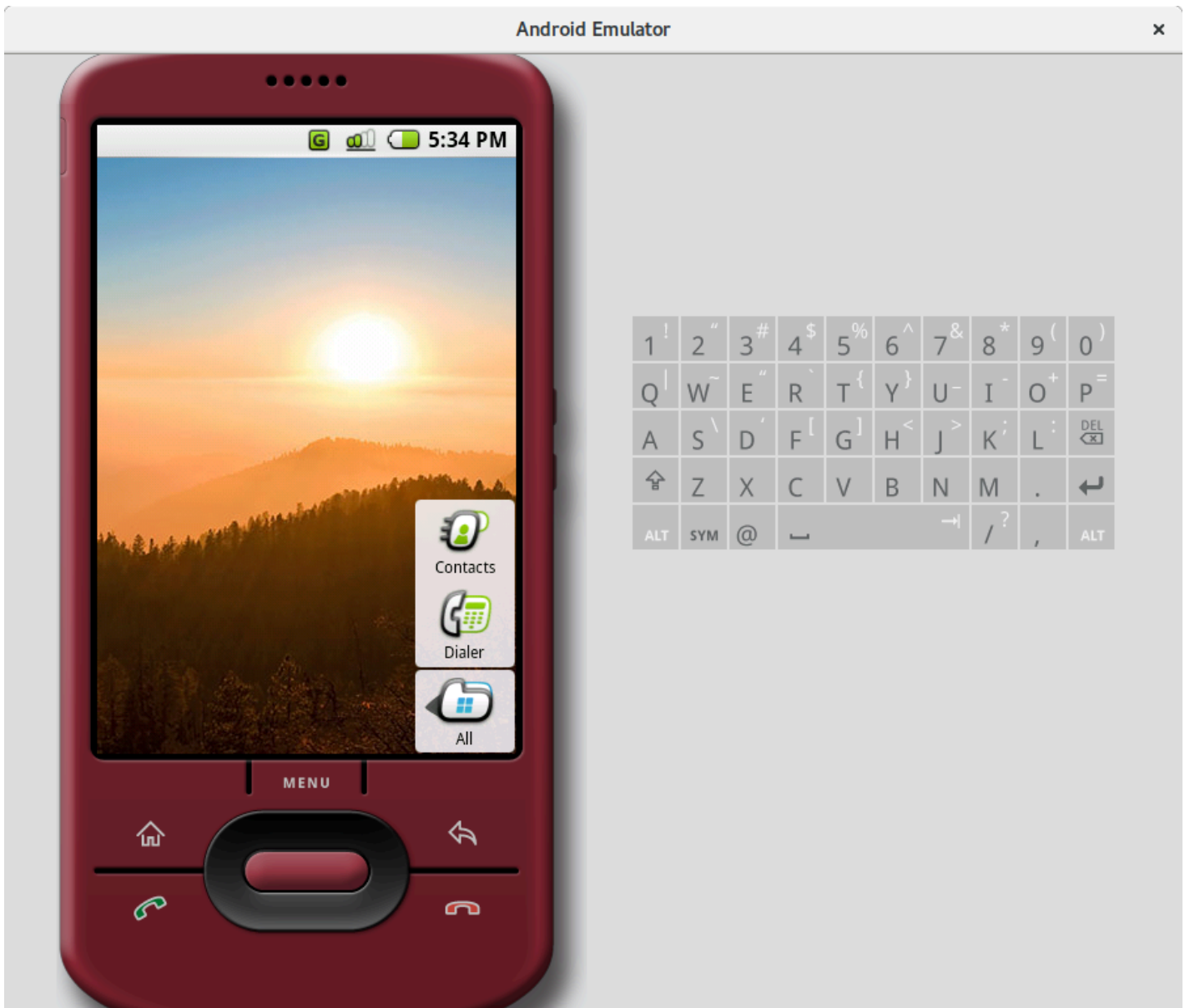
Android m3 и m5

Немногие знают, но первой публичной версией Android был вовсе не Android 1.0, выпущенный в 2008 году, когда уже существовала iPhone OS 2.0, а версия под названием m3, выпущенная почти одновременно с первым iPhone.



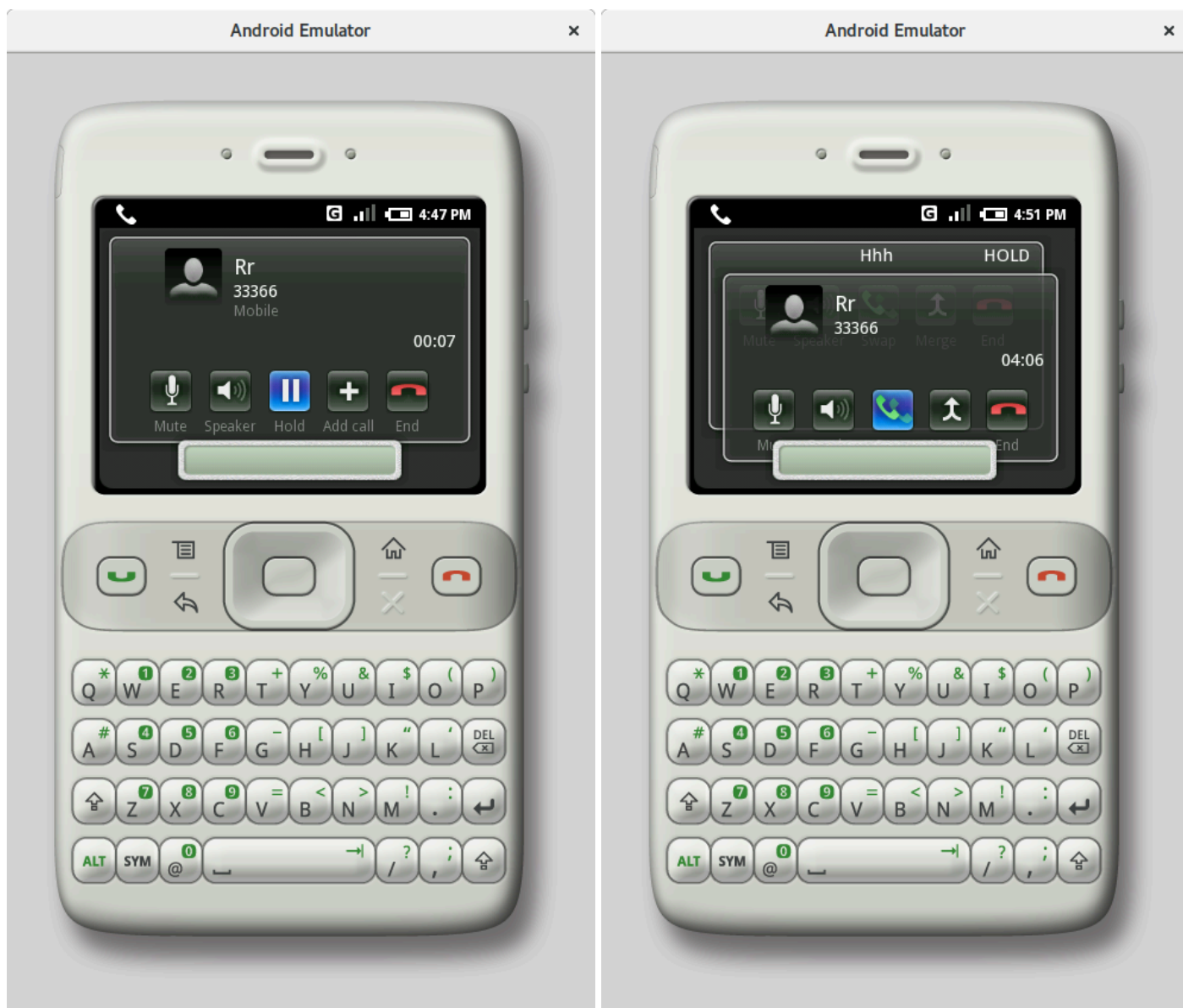
Android m3

Этот эмулятор изображает HTC Sooner (он же HTC EXCA 300), который должен был стать первым смартфоном на Android. Видно, что Google метила в бизнес-сегмент пользователей: владельцы BlackBerry, Nokia серии E, Samsung серии i (i780). После выхода iPhone позиция Google резко поменялась, HTC Sooner был отменен, а в эмуляторе Android m5, выпущенном всего через несколько месяцев, изображение смартфона стало более привычным.



Android m5

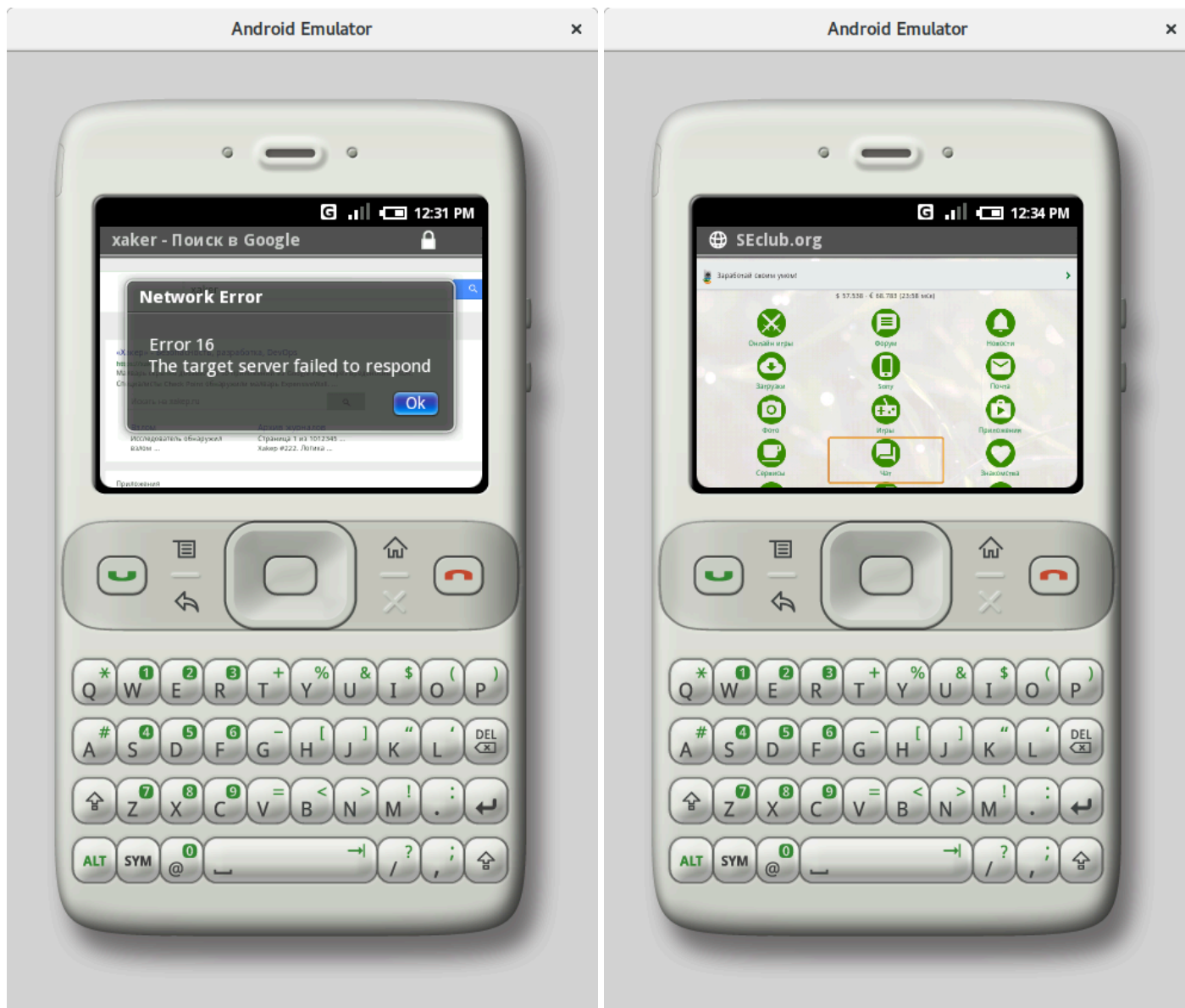
Несмотря на наличие клавиатуры и джойстика Android m3 поддерживал сенсорное управление. На любой элемент можно было нажать, имелась поддержка свайпов (на главном экране свайпаем вправо и влево по панели приложений, тем самым прокручивая менюшку с этими самими приложениями). Поддержка прокрутки списков жестом тоже была: можно, например, прокрутить веб-страницу в браузере.



Исходящий вызов и группой разговор (m3)

Вот только сенсорный экран никак не упрощал пользование устройством. Он был как будто для галочки. Элементы интерфейса слишком мелкие и не рассчитаны на управление пальцем. Совсем другое дело — версия m5: элементы интерфейса большие. По ним сложно промахнуться пальцем. Лаунчер выглядит хоть и необычно, но уже более знакомо (см. скриншот): справа отображаются четыре последних запущенных приложения.

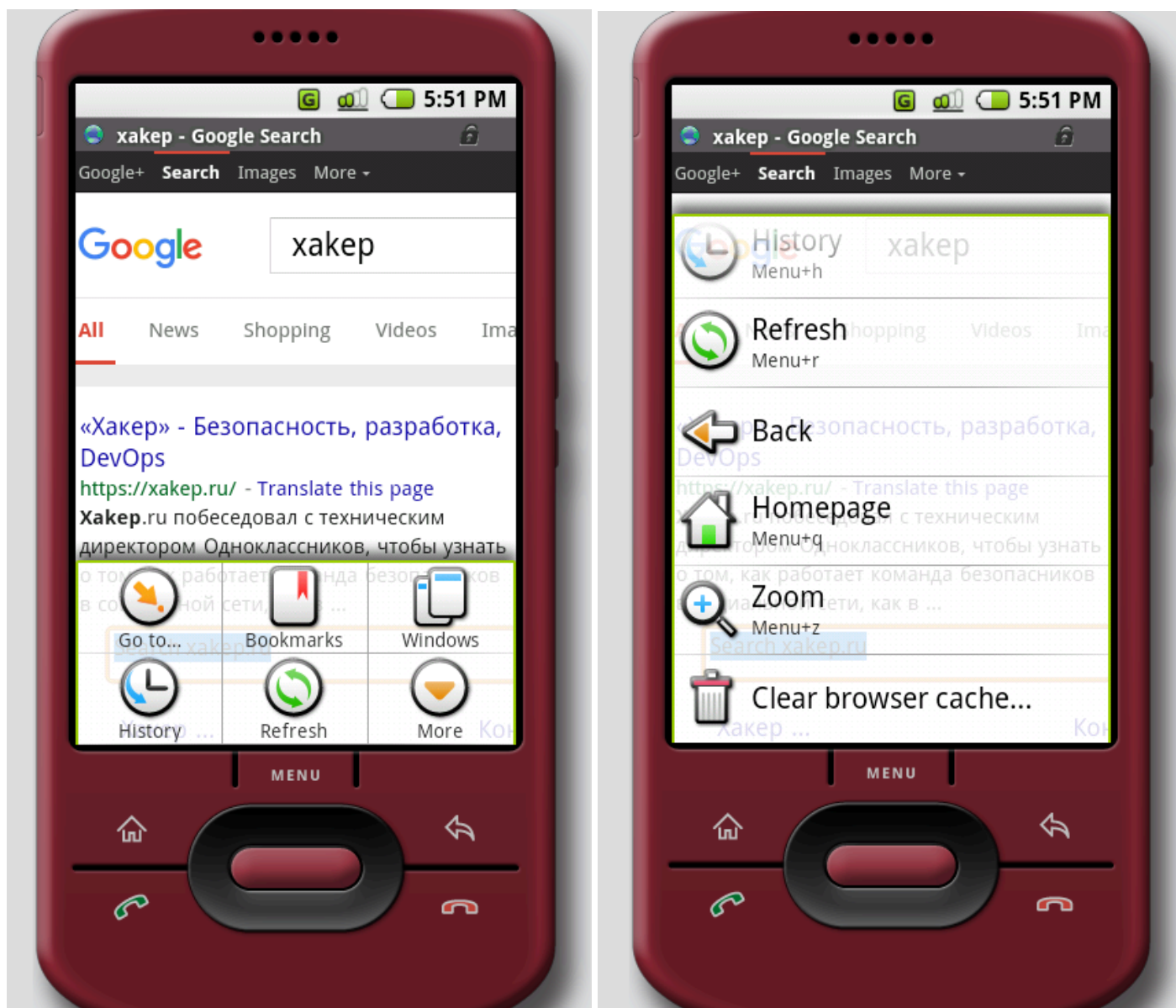
Как и в iPhone OS, в Android m3 был полноценный браузер. Сегодня он, конечно же, почти не работает, все время говорит о просроченных сертификатах, жутко тормозит, многие сайты либо не отображает, либо показывает с искажениями и не загруженными элементами.



Хакер.ru не открывается, но WAP работает

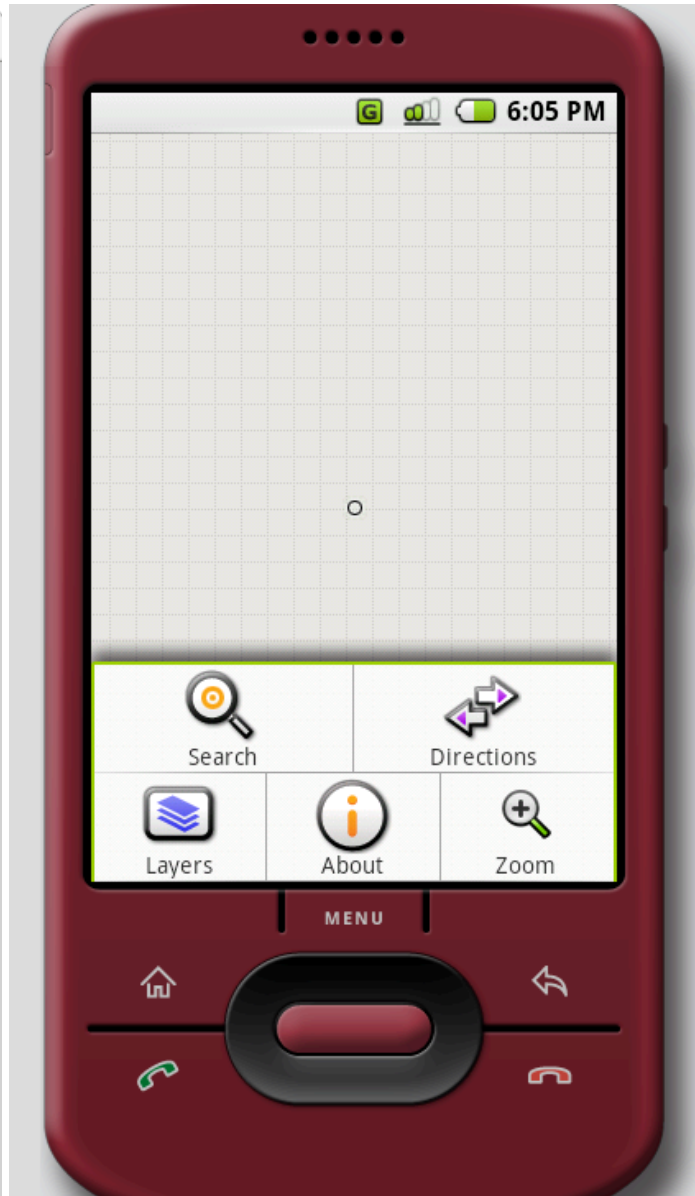
Впрочем, браузер не оставляет впечатления сырого софта. Он даже не хуже, чем стандартные браузеры, которые ставили на обычные телефоны того времени. Тут есть и закладки с историей, и функция отдельной очистки истории, кэша, Cookies.

В m5 браузер уже другой, и заточен под сенсорное управление, однако на этом отличия заканчиваются.



Браузер в m5

В обеих версиях есть предустановленные карты Google. Только с помощью этого приложения сами карты не удастся — они не загружаются. К счастью, даже без них можно оценить интерфейс и функциональность приложения.



Вот что значит простые карты (слева m3, справа — m5)

Красота! Не правда ли? Даже в версии «Гуглокарт» на Java было больше функций (да, да, Google выпускала свои карты для телефонов с Java; они умели определять местоположение по сотовым вышкам и GPS и даже умели строить маршруты).



Контакты и последние вызовы (m3)

Заметно, что m3 имеет больше общего с обычными кнопочными телефонами и на iOS он абсолютно ничем не похож. Зато теперь мы знаем, откуда в старых смартфонах с Android есть кнопка «Меню» — это аналог правой клавиши (или клавиши «опции») на обычном телефоне.

В свое время Android m3 мог бы стать отличным конкурентом Symbian и кнопочным телефонам. Сравнить его с Android 8 не имеет смысла. Сравнить Android m5 с новым Android 8 тоже не имеет смысла, это явно концепт, который в Google в спешке выкатили после релиза iPhone.

Краткая история версий

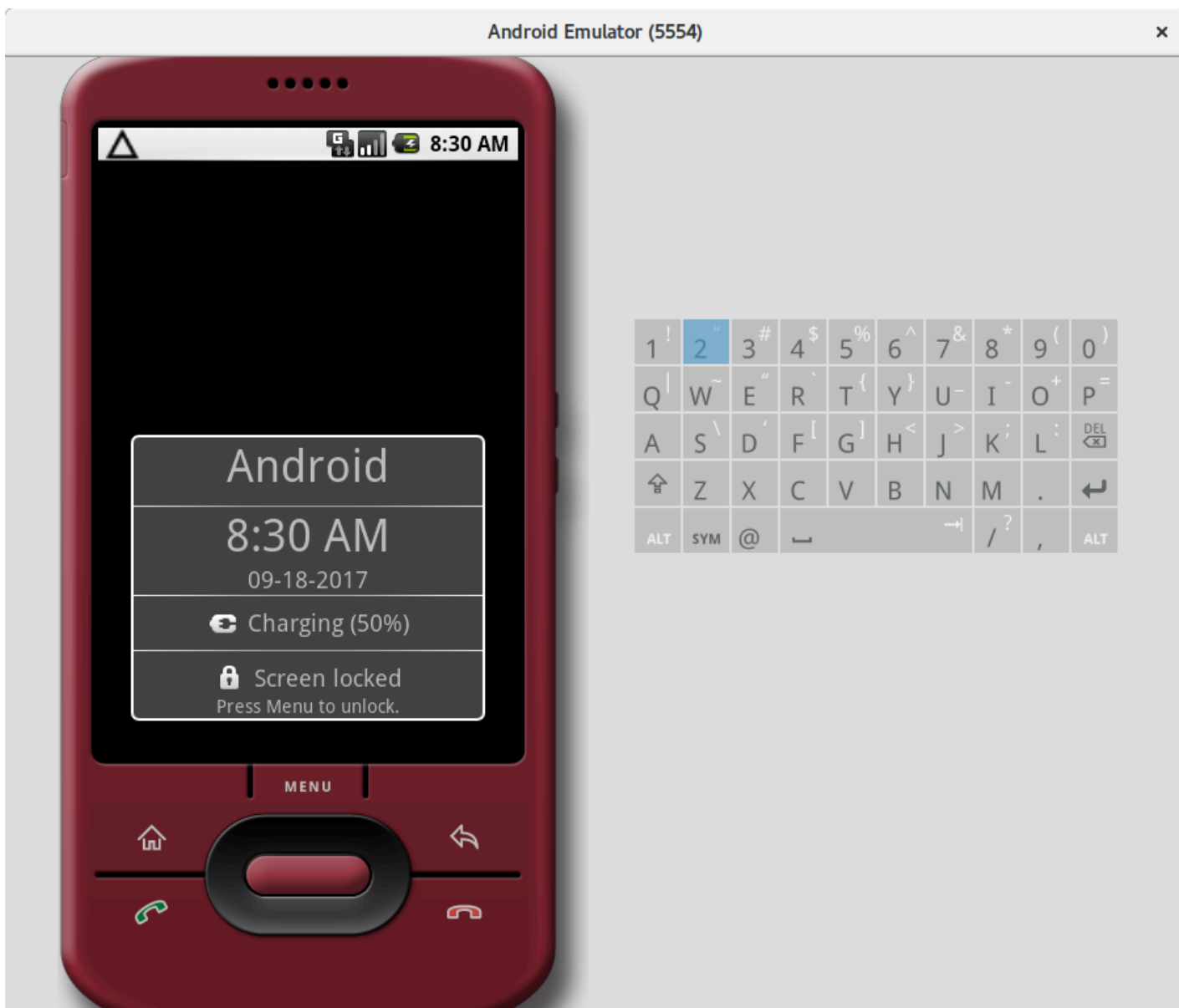
- Android m3 — 12 ноября 2007 года. Версия API: неизвестно. Версия ядра Linux: неизвестно. Это первая бета-версия, доступная публично.

- Android m5 — 13 февраля 2008 года. Версия ядра Linux: 2.6.23. Версия API: неизвестно.
- Android 0.9 — 18 августа 2008 года. Версия ядра Linux: 2.6.25. Версия API: неизвестно. Это релиз-кандидат. Почти стабильная версия.
- Android 1.0 — 23 сентября 2008 года. Версия API: 1. Версия ядра 2.6.25. Первая стабильная версия, то есть релиз.
- Android 1.1 — 9 февраля 2009 года. Версия API: 2. Версия ядра 2.6.25.
- Android 1.5 — 30 апреля 2009 года. Версия API: 3. Версия ядра Linux: 2.6.27.
- Android 1.6 — 15 сентября 2009 года. Версия API: 4. Версия ядра Linux: 2.6.29.

Первый смартфон на ОС Android назывался HTC Dream (также известный как T-Mobile G1). Он работал под управлением Android 1.0 (первая сборка TC4-RC19, датируемая 13 сентября 2008 года) и был выпущен в продажу 31 октября 2008 года. В скором времени он получил обновление до Android 1.1 и далее обновлялся вплоть до Android 1.6.

Android 1.0–1.5

После загрузки видим экран блокировки. По современным меркам он выглядит ужасно, но почти не уступает экрану из Android Oreo. Разве что пропущенных событий и уведомлений не хватает. Для разблокировки устройства нужно нажать кнопку «Меню». Никакие свайпы тут не работают.



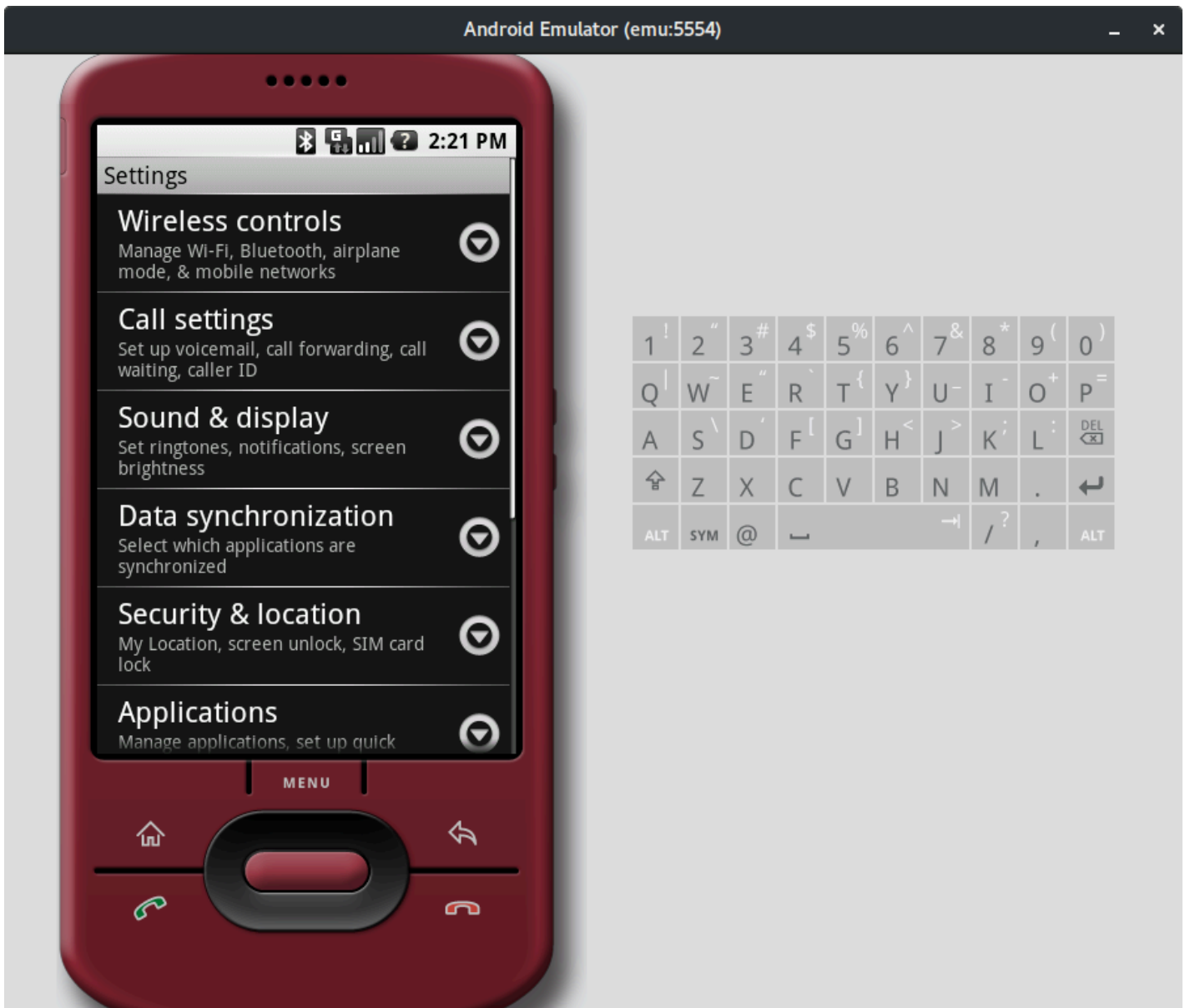
Экран блокировки

Лаунчер выглядит знакомо. Он принял вид, в котором будет оставаться вплоть до версии Android 1.6. Он имеет поддержку виджетов и ярлыков, которые можно свободно перемещать по рабочему столу. В целом концепция такая же, как и в Pixel Launcher, который установлен по умолчанию в Android 7+. За тем исключением, что он не умеет автоматически перемещать ярлыки на свободное место и не позволяет менять размер виджетов.



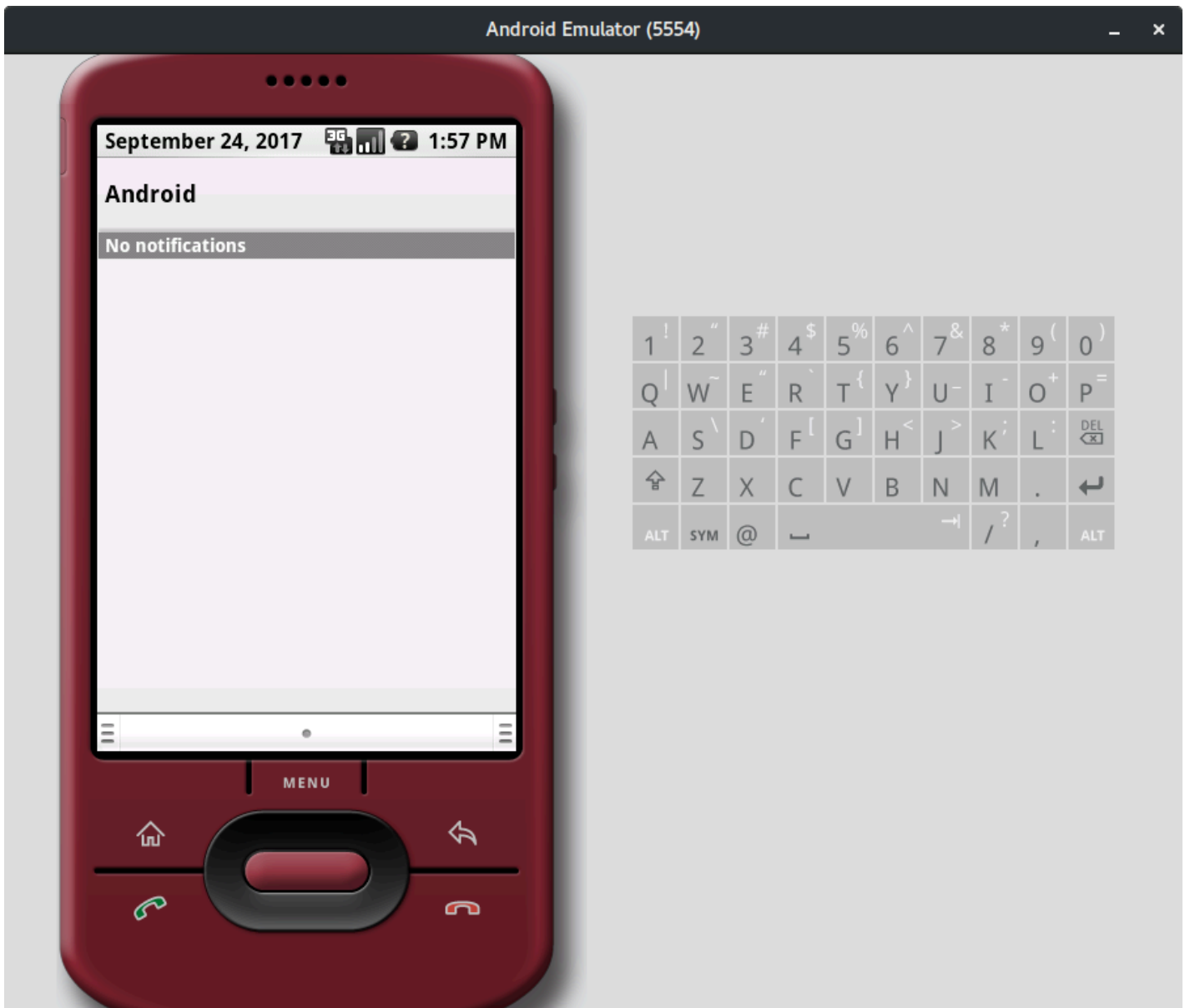
Домашний экран

Как ни странно, в 1.0 не было настроек, они появились только в 1.5.



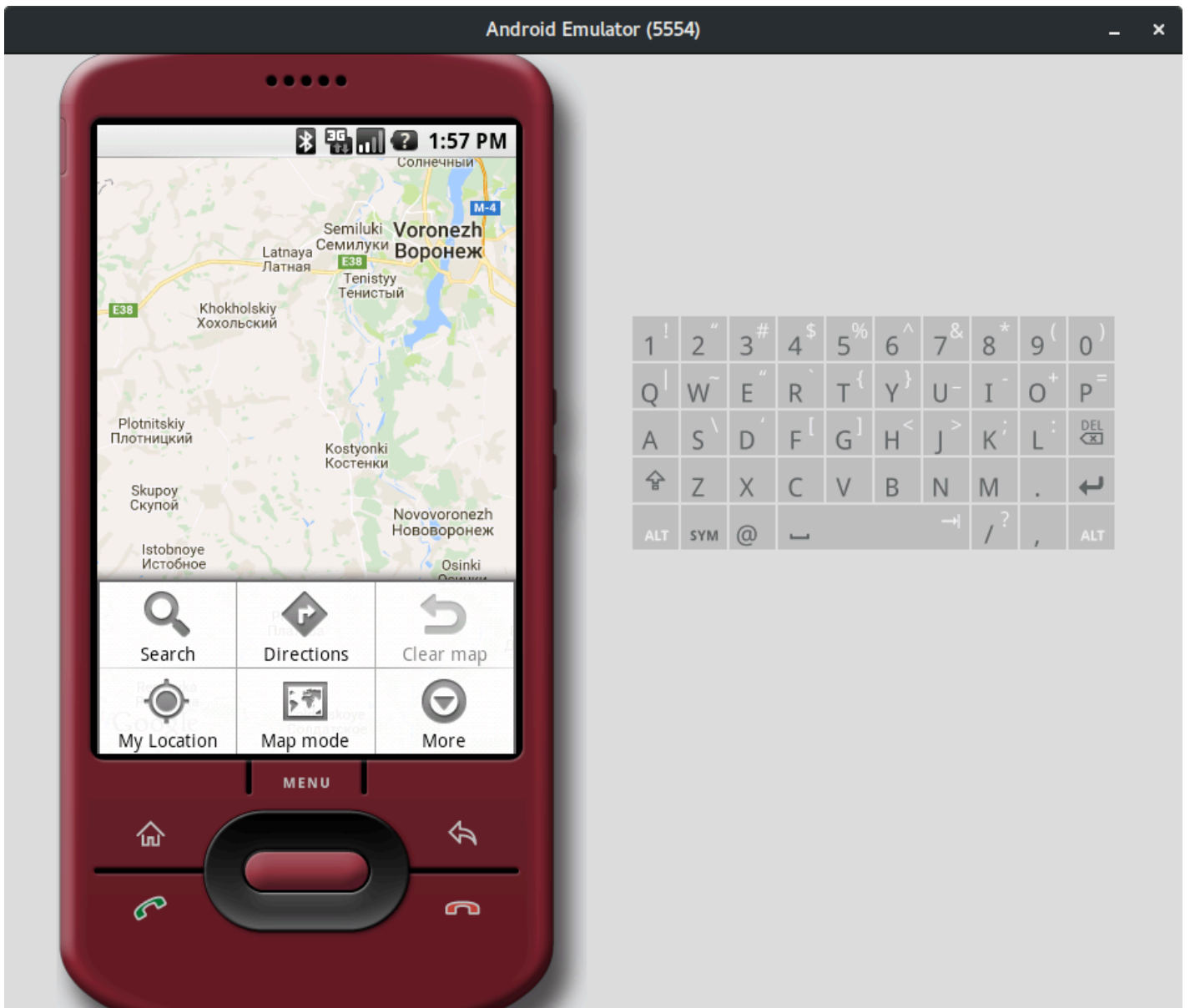
Настройки в Android 1.5

Панель уведомлений присутствует, но в сравнении с современными версиями ее функции сводятся только к двум вещам: возможность нажать на уведомление, чтобы открыть приложение и возможность смахнуть уведомление. Никакой панели быстрых настроек, никаких настроек показа уведомления и их блокировки. Уведомления не могут разворачиваться и содержать кнопки. И, конечно же, здесь нет позаимствованной из iOS функции Heads Up, которая выводит плашку с уведомлением в верхней части экрана.



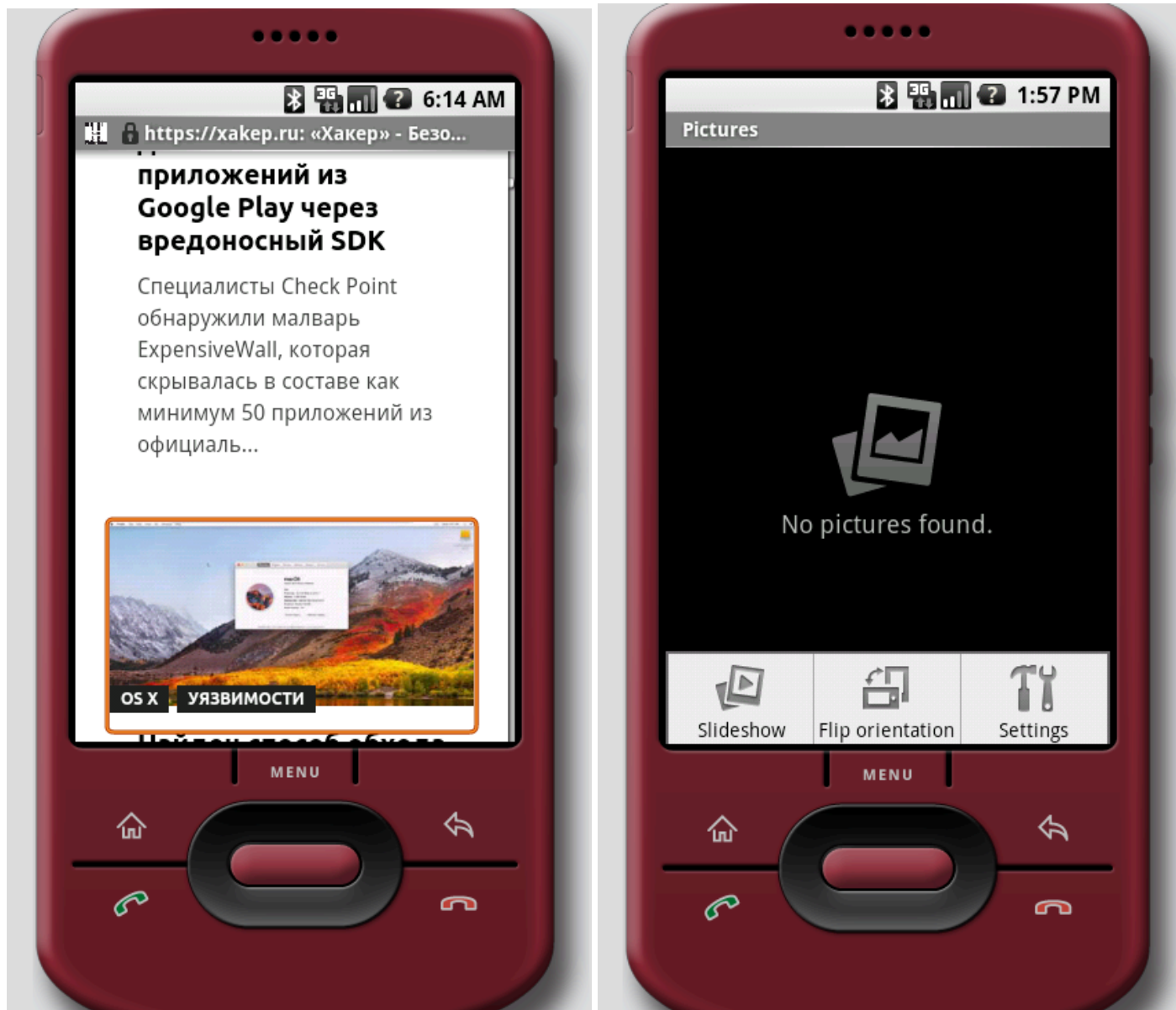
Панель уведомлений

Карты в 1.0 до сих пор работают. Они умеют показывать объекты (например, магазины), однако не показывают дополнительные данные, такие как телефон, часы работы, посещаемость и т.д. Нет функции быстрого поиска поблизости нужных заведений (больниц, кафе, баров, заправок, супермаркетов и т. д.). Зато по сей день работает режим просмотра улиц. Поддержки мультитача нет, да и вообще пользоваться крайне неудобно.

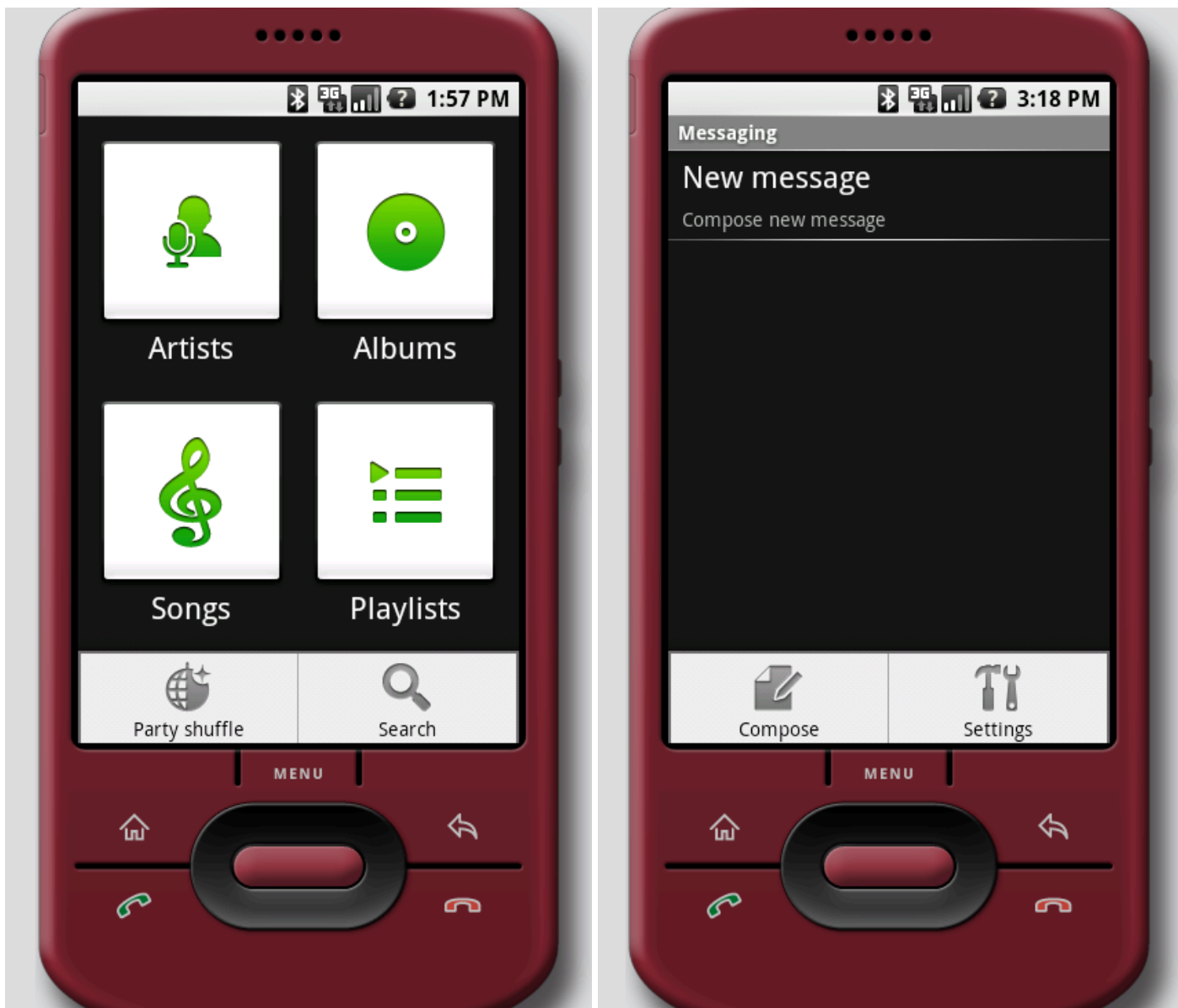


Карты

В браузере появилась поддержка вкладок! Очень важные изменения произошли и внутри: создатели явно обновили движок браузера. Теперь он корректно распознается сайтами как мобильный браузер (да, в m5 и ниже браузер пытался загрузить тяжеловесную страницу для ПК) и работает куда быстрее. Удалось открыть главную страницу нашего журнала. При попытке открыть статью браузер вылетел. Впрочем, как я позже выяснил, браузер любит вылетать при открытии тяжеловесных сайтов с кучей скриптов и графики.



Браузер (слева), галерея (справа)



Музыкальный плеер (слева), сообщения (справа)

Как менялся интерфейс Android

1. m3 — версия для кнопочных смартфонов без сенсорного экрана. Элементы интерфейса мелкие, преобладают черный и синий цвета. Присутствует множество эффектов анимации, меню и многие другие элементы полупрозрачные. Кнопки в стиле старой macOS.
2. m5 — интерфейс полностью переделан под сенсорные экраны, элементы управления стали намного крупнее. Многие эффекты анимации пропали. В интерфейсе преобладает белый цвет.
3. 3.0 — новая тема оформления Holo (от Holographic): темный фон, яркие элементы управления с прямыми линиями и эффектом свечения. Экранные клавиши навигации вместо аппаратных.

4. 4.0 — тот же Holo без эффекта свечения. Темный фон преобладает, но также используется белый.
5. 5.0 — Material Design, переживавший также и в другие продукты Google. Яркий плоский интерфейс с тенями и сложными эффектами анимации.

Технические аспекты

Если судить по начинке HTC Dream, рекомендуемые системные требования Android 1.0 были примерно такими:

- наличие сенсорного экрана с разрешением 320x480;
- наличие клавиш приема и отбоя вызова, кнопок «Меню», «Домой», «Назад»;
- процессор 528 МГц;
- ОЗУ 192 Мбайт;
- ПЗУ 256 Мбайт.

Однако наше небольшое расследование подтверждает (см. скриншоты) — ОС вполне могла бы уместиться в 128 Мбайт ПЗУ. На эмулируемом устройстве под системный раздел отведено 64 мегабайта, из которых системой занято 42,3 мегабайт, а под будущие обновления оставлено 23,2 мегабайта. В

```
/data
```

(данные программ и сторонние приложения) из 64 мегабайт занято 29,8, а свободно 35,7. Под кеш выделено тоже 64 мегабайта, однако занимает он только 1,1. В итоге: система занимает 73 мегабайта а остальное можно оставить под сторонние приложения, которые в то время не часто весили больше мегабайта.

```
# df
/dev: 47284K total, 0K used, 47284K available (block size 4096)
/sqlite_stmt_journals: 4096K total, 0K used, 4096K available (block size 4096)
/system: 65536K total, 42320K used, 23216K available (block size 4096)
/data: 65536K total, 29848K used, 35688K available (block size 4096)
/cache: 65536K total, 1156K used, 64380K available (block size 4096)
# █
```

Распределение ПЗУ

Достаточный для работы ОС объем ОЗУ — 64 мегабайта, во-первых, потому, что в эмуляторе система прекрасно обходится 92-мя, а во-вторых, в старые времена китайцы клепали смартфоны с 64 мегабайтами и они прекрасно работали (как владелец одного из таких, подтверждаю — прим. ред.).

```
# cat meminfo
MemTotal:      94572 kB
MemFree:       12076 kB
Buffers:        0 kB
Cached:        44064 kB
SwapCached:    0 kB
Active:        46712 kB
Inactive:      27900 kB
SwapTotal:     0 kB
SwapFree:      0 kB
Dirty:         0 kB
Writeback:     0 kB
AnonPages:    30572 kB
Mapped:        20304 kB
Slab:          2976 kB
SReclaimable:  616 kB
SUnreclaim:   2360 kB
PageTables:   2464 kB
NFS_Unstable:  0 kB
Bounce:       0 kB
CommitLimit:  47284 kB
Committed_AS: 874072 kB
VmallocTotal: 876544 kB
VmallocUsed:  10436 kB
VmallocChunk: 855036 kB
#
```

Распределение ОЗУ

Посмотрим, что представлял собой образ с системой. Для этого распакуем `system.img`

. Будучи сжатым, этот образ занимает всего 26 Мбайт, а после распаковки — 41,3 Мбайт. По сегодняшним меркам, когда обычное приложение может занимать 100 мегабайт, это просто копейки. Однако это полноценная операционная система, которая умеет многое из того, что умеет современный Android.

Имя	Размер	Дата изменения
app	26 объектов	23 сен. 2008
bin	94 объекта	23 сен. 2008
build.prop	1,1 кБ	23 сен. 2008
etc	14 объектов	23 сен. 2008
fonts	8 объектов	23 сен. 2008
framework	18 объектов	23 сен. 2008
lib	61 объект	23 сен. 2008
media	1 объект	23 сен. 2008
sounds	1 объект	12:43
usr	4 объекта	23 сен. 2008
xbin	3 объекта	23 сен. 2008

Имя	Размер	Дата изменения
app	61 объект	1 янв. 2009
bin	311 объектов	1 янв. 2009
build.prop	1,9 кБ	1 янв. 2009
compatibility_matrix.xml	10,3 кБ	1 янв. 2009
etc	33 объекта	1 янв. 2009
fake-libs	1 объект	1 янв. 2009
fake-libs64	1 объект	1 янв. 2009
fonts	171 объект	1 янв. 2009
framework	55 объектов	1 янв. 2009
lib	291 объект	1 янв. 2009
lib64	341 объект	1 янв. 2009
manifest.xml	2,5 кБ	1 янв. 2009
media	2 объекта	1 янв. 2009
priv-app	79 объектов	1 янв. 2009
usr	7 объектов	1 янв. 2009
vendor	7 байт	11:28

Структура каталогов Android 1.0 (слева) и 7.1 (справа)

Все системные приложения и сервисы находятся в каталоге

`app`

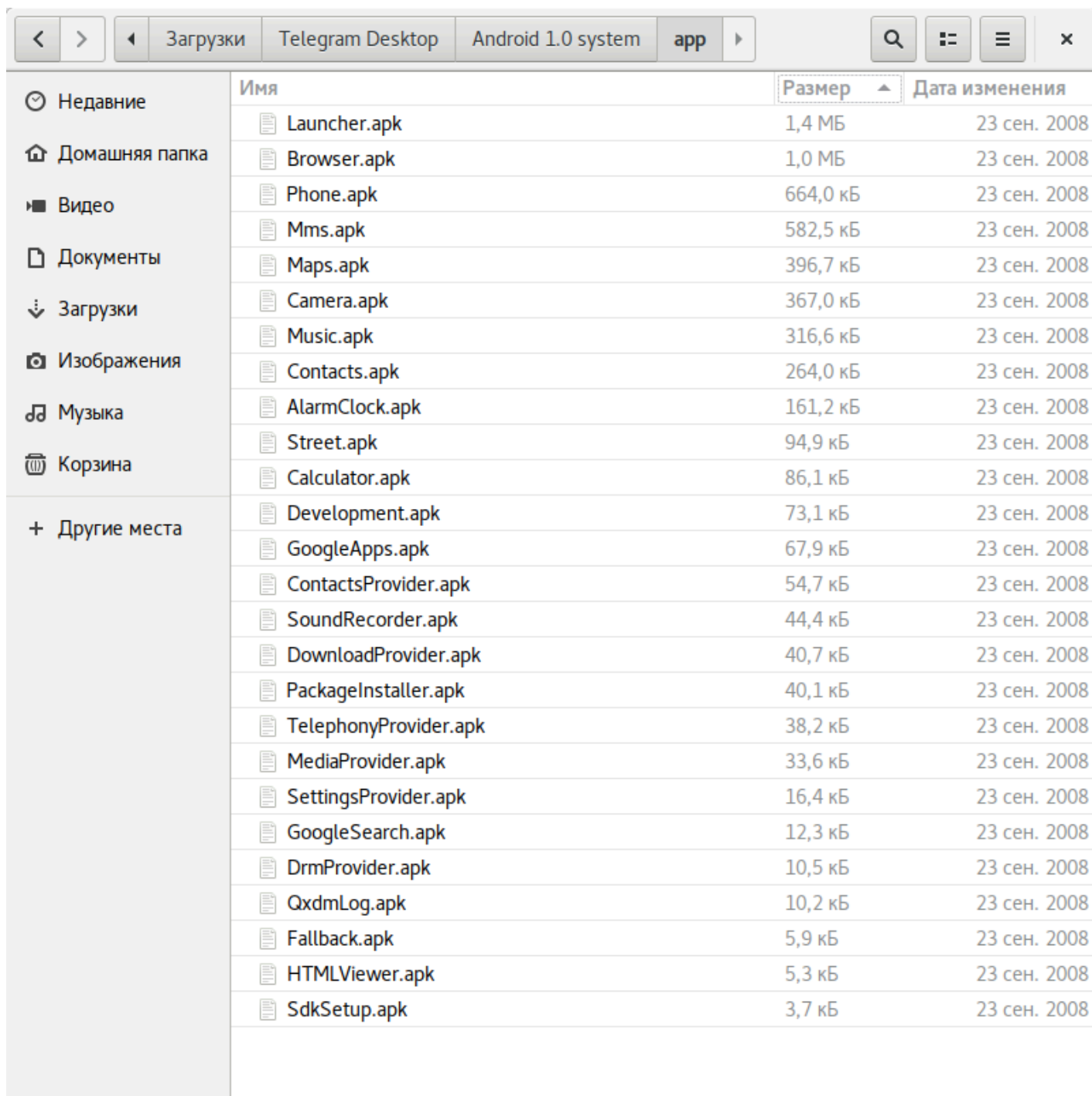
. В современном Android приложения с особыми привилегиями (в основном это системные сервисы), вынесены в отдельный каталог

`priv-app`

Открываем

`app`

и видим великое чудо: программы занимают ничтожно мало места. Самые жирные: лаунчер (1,32 Мбайт) и браузер (0,98 Мбайт). Встречаются приложения размером меньше 100 Кбайт (диктофон — 44,4 Кбайт, калькулятор — 86,1 Кбайт, просмотр улиц — 94,9 Кбайт).



Имя	Размер	Дата изменения
Launcher.apk	1,4 МБ	23 сен. 2008
Browser.apk	1,0 МБ	23 сен. 2008
Phone.apk	664,0 кБ	23 сен. 2008
Mms.apk	582,5 кБ	23 сен. 2008
Maps.apk	396,7 кБ	23 сен. 2008
Camera.apk	367,0 кБ	23 сен. 2008
Music.apk	316,6 кБ	23 сен. 2008
Contacts.apk	264,0 кБ	23 сен. 2008
AlarmClock.apk	161,2 кБ	23 сен. 2008
Street.apk	94,9 кБ	23 сен. 2008
Calculator.apk	86,1 кБ	23 сен. 2008
Development.apk	73,1 кБ	23 сен. 2008
GoogleApps.apk	67,9 кБ	23 сен. 2008
ContactsProvider.apk	54,7 кБ	23 сен. 2008
SoundRecorder.apk	44,4 кБ	23 сен. 2008
DownloadProvider.apk	40,7 кБ	23 сен. 2008
PackageInstaller.apk	40,1 кБ	23 сен. 2008
TelephonyProvider.apk	38,2 кБ	23 сен. 2008
MediaProvider.apk	33,6 кБ	23 сен. 2008
SettingsProvider.apk	16,4 кБ	23 сен. 2008
GoogleSearch.apk	12,3 кБ	23 сен. 2008
DrmProvider.apk	10,5 кБ	23 сен. 2008
QxdmLog.apk	10,2 кБ	23 сен. 2008
Fallback.apk	5,9 кБ	23 сен. 2008
HTMLViewer.apk	5,3 кБ	23 сен. 2008
SdkSetup.apk	3,7 кБ	23 сен. 2008

Это все приложения Android 1.0

Заглянем в папку

etc

. Тут хранятся конфигурационные файлы (настройки) различных системных утилит.

Интерес представляет каталог

otacerts

. Судя по названию и файлам, содержащимся в нем, Android 1.0 поддерживает OTA-обновления.

Каталога lib64 нет, ОС не поддерживает 64-битные приложения. Библиотеки Java-классов (framework) занимают всего 6,3 Мбайт (в сравнении с 251,3 Мбайт в современной версии).

Выполняем команду

```
mount
```

и узнаем, что Android изначально использовал файловую систему yaffs2, предназначенную для работы на флешках с высоким уровнем износа (в то время качество NAND памяти оставляло желать лучшего, часто появлялись битые сектора). Yaffs была однопоточной файловой системой с низкой производительностью, поэтому в версии 2.3 программисты Google ее выпилили, заменив на классическую для систем на ядре Linux ext4.

Еще одна особенность Android 1.0 — отсутствие JIT-компилятора. Официальным языком Android всегда был Java, но в качестве виртуальной машины для исполнения приложений здесь использовалась более эффективная в плане потребления памяти регистровая VM Dalvik (в противовес стековой виртуальной машины Java). Она хоть и была менее требовательной к оперативной памяти, но серьезно проигрывала обычной Java в скорости исполнения приложений.

Google исправила это только в версии 2.2, реализовав, наконец, JIT-компилятор, который вместо интерпретации байткода на лету переводил его в машинные инструкции. В Android 5.0 ему на смену пришел AOT-компилятор, который конвертировал программу в машинные инструкции при установке, замедляя таким образом процесс инсталляции и лишая виртуальную машину многих возможностей оптимизации.

Чтобы это исправить, в Android 7 решили использовать гибридный JIT/AOT-компилятор. Сразу после установки приложение исполняется с использованием JIT-компилятора, но во время простоя смартфона на зарядке подключается AOT-компилятор и перегоняет приложение в машинные инструкции с помощью AOT-компилятора (на самом деле все намного сложнее, но для объяснения нюансов потребовалась бы отдельная статья). Интересно, что в Android 1.0 уже были песочницы для приложений. И реализованы они были весьма интересным образом. Каждое приложение запускалось от имени отдельного пользователя Linux и, таким образом, имело доступ только к своему каталогу внутри

```
/data/data
```

```

# ls -l data/data
drwxr-xr-x app_0    app_0    2017-09-10 19:01 com.android.alarmclock
drwxr-xr-x app_1    app_1    2017-09-10 19:01 com.android.calculator2
drwxr-xr-x app_2    app_2    2017-09-10 19:01 com.android.contacts
drwxr-xr-x app_3    app_3    2017-09-10 19:01 com.android.camera
drwxr-xr-x app_3    app_3    2017-09-10 19:01 com.android.providers.downloads
drwxr-xr-x app_2    app_2    2017-09-10 19:01 com.android.providers.contacts
drwxr-xr-x app_4    app_4    2017-09-10 19:01 com.android.fallback
drwxr-xr-x app_3    app_3    2017-09-10 19:01 com.android.providers.drm
drwxr-xr-x app_2    app_2    2017-09-10 19:01 com.android.googlesearch
drwxr-xr-x app_5    app_5    2017-09-10 19:01 com.android.htmlviewer
drwxr-xr-x app_2    app_2    2017-09-10 19:01 com.android.launcher
drwxr-xr-x app_3    app_3    2017-09-10 19:01 com.android.providers.media
drwxr-xr-x app_6    app_6    2017-09-10 19:01 com.android.music
drwxr-xr-x app_7    app_7    2017-09-10 19:01 com.android.packageinstaller
drwxr-xr-x radio    radio    2017-09-10 19:01 com.android.phone
drwxr-xr-x app_8    app_8    2017-09-10 19:01 com.android.mms
drwxr-xr-x system  system  2017-09-10 19:01 com.android.qxdmlog
drwxr-xr-x app_9    app_9    2017-09-10 19:01 com.android.sdksetup
drwxr-xr-x app_10   app_10  2017-09-10 19:01 com.android.soundrecorder
drwxr-xr-x system  system  2017-09-10 19:01 com.android.providers.settings
drwxr-xr-x radio    radio    2017-09-10 19:01 com.android.providers.telephony
drwxr-xr-x app_11   app_11  2017-09-10 19:01 com.google.android.street
drwxr-xr-x app_12   app_12  2017-09-10 19:01 com.android.browser
drwxr-xr-x app_13   app_13  2017-09-10 19:01 com.android.development
drwxr-xr-x app_14   app_14  2017-09-10 19:01 com.google.android.googleapps
drwxr-xr-x app_15   app_15  2017-09-10 19:01 com.google.android.apps.maps
drwxr-xr-x app_16   app_16  2017-09-10 19:01 com.example.android.apis
drwx----- system  system  2017-09-10 19:01 com.android.settings
#

```

Каталоги приложений принадлежат разным пользователям и доступны для чтения и модификации только им

Друг с другом и с операционной системой приложения могли общаться только через IPC-механизм Binder, который требовал авторизации на выполнение действий. Этот же механизм использовался и для несколько других целей: с его помощью система оповещала приложения о системных событиях, таких как входящий вызов, пришедшее SMS, втыкание зарядки и так далее. Приложения получали сообщения и могли на них отреагировать.

Благодаря такой особенности в Android были широкие возможности автоматизации, о которых мы знаем благодаря таким приложениям как Tasker, Automate или Locale. Все эти приложения доступны и для Android 8, разве что некоторые опасные возможности, такие как включение и выключение режима полета, теперь запрещены для использования обычными приложениями.

В Android 1.0 не было поддержки push-уведомлений и никаких ограничений на работу приложений в фоне. Последним часто пользовались криворукие разработчики, создававшие приложения, которые быстро выжирали заряд аккумулятора. Google исправила эту проблему только в Android 8, запретив запуск фоновых служб (правда, до нее это же успели сделать разработчики китайских прошивок вроде MIUI).

Ну и самое главное — первый Android тормозил. И тормозил он вплоть до версии 3.0, которая наконец научилась использовать графический процессор для обработки 2D-графики. Почему такой функции не было изначально? Просто потому, что Android должен был работать на всем, включая самые дешевые смартфоны, в которые в то время графический ускоритель не предустанавливали.

Парадокс новизны и безопасности

Год от года разработчики Google добавляли в Android новые механизмы защиты. Поэтому на первый взгляд может показаться, что Android 1.0 и 8.0 это как Windows 95 и 10 в плане безопасности. Но это не совсем так. Как мы уже выяснили, в дизайне ОС изначально существовали песочницы для приложений, IPC-механизм и система разрешений с многоуровневой проверкой полномочий (вплоть до ядра Linux). Большая часть ОС была написана на языке Java, который в силу своего дизайна не позволял писать код, в котором можно вызвать переполнение буфера, выйти за границы массива, повторно использовать освобожденную память и применять другие приемы взлома. Однако со временем операционная система обростала новыми функциями и возможностями. А чем сложнее и длиннее код, тем труднее его отлаживать и исправлять ошибки. По мере развития системы появлялись целые подсистемы, пригодные для эксплуатации злоумышленниками.

Например, в 1.0 не было мультимедиа-библиотеки stagefright (она появилась в Android 2.3), в которой нашли [массу уязвимостей](#), позволяющих взломать смартфон, отправив на него MMS или попросив пользователя открыть специальную ссылку.

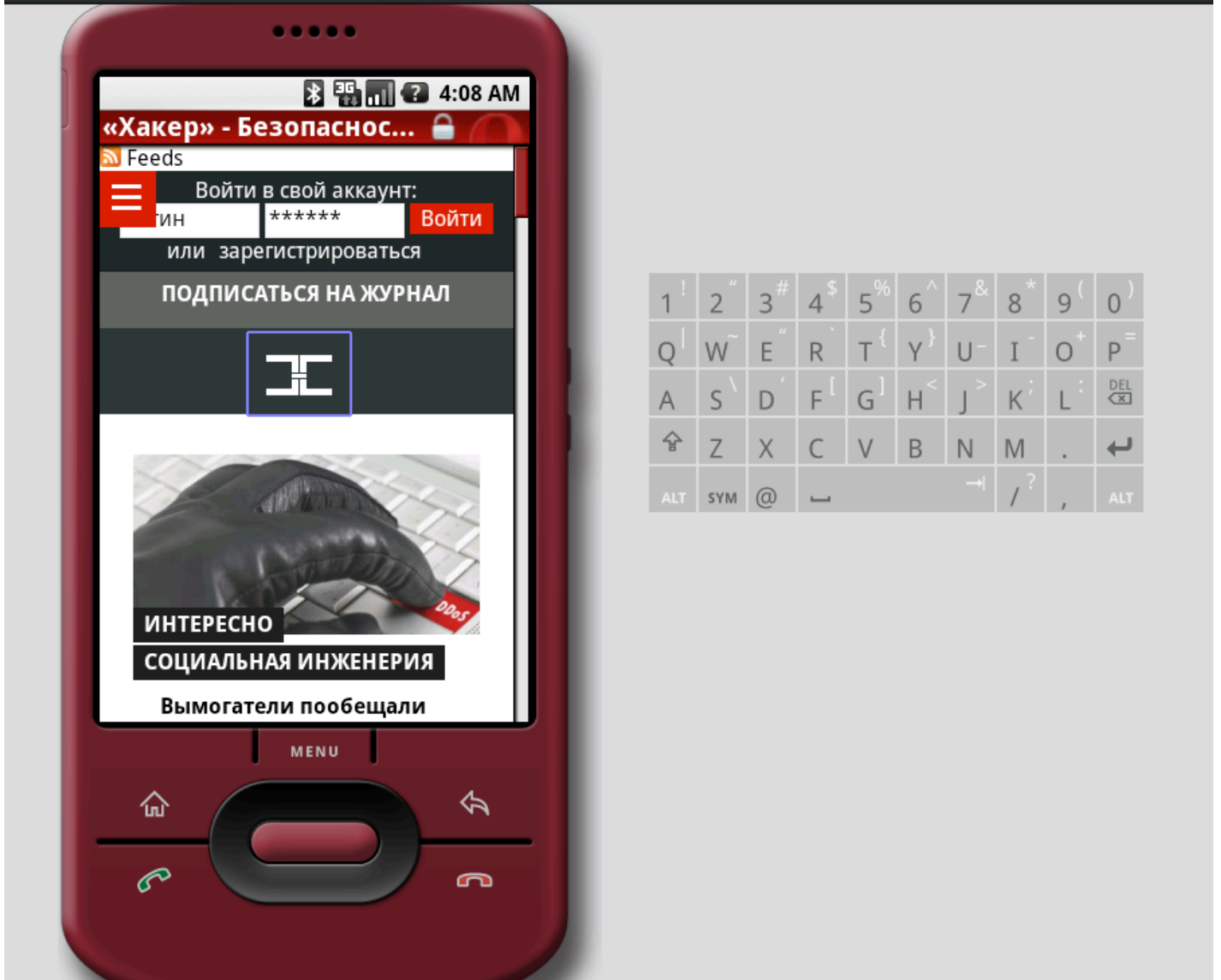
В Android 1.0 не было и сервиса Accessibility («Специальные возможности»), с помощью которого можно получить полный контроль над смартфоном. Не было также режима администратора, который так любят использовать вирусы, чтобы защититься от удаления.

С другой стороны, тогда не существовало многих внедренных позже средств защиты: [SELinux](#), [Seccomp](#), системы запроса и управления полномочиями приложений, механизма проверки целостности системы Verified Boot, который появился в Android 4.4, но полноценно начал работать только в 7.0, шифрования данных, защиты от даунгрейда, онлайн-антивируса Verify Apps, работающего со времен 4.2. На самом деле ты даже PIN-код не мог установить на экран.

Старый софт

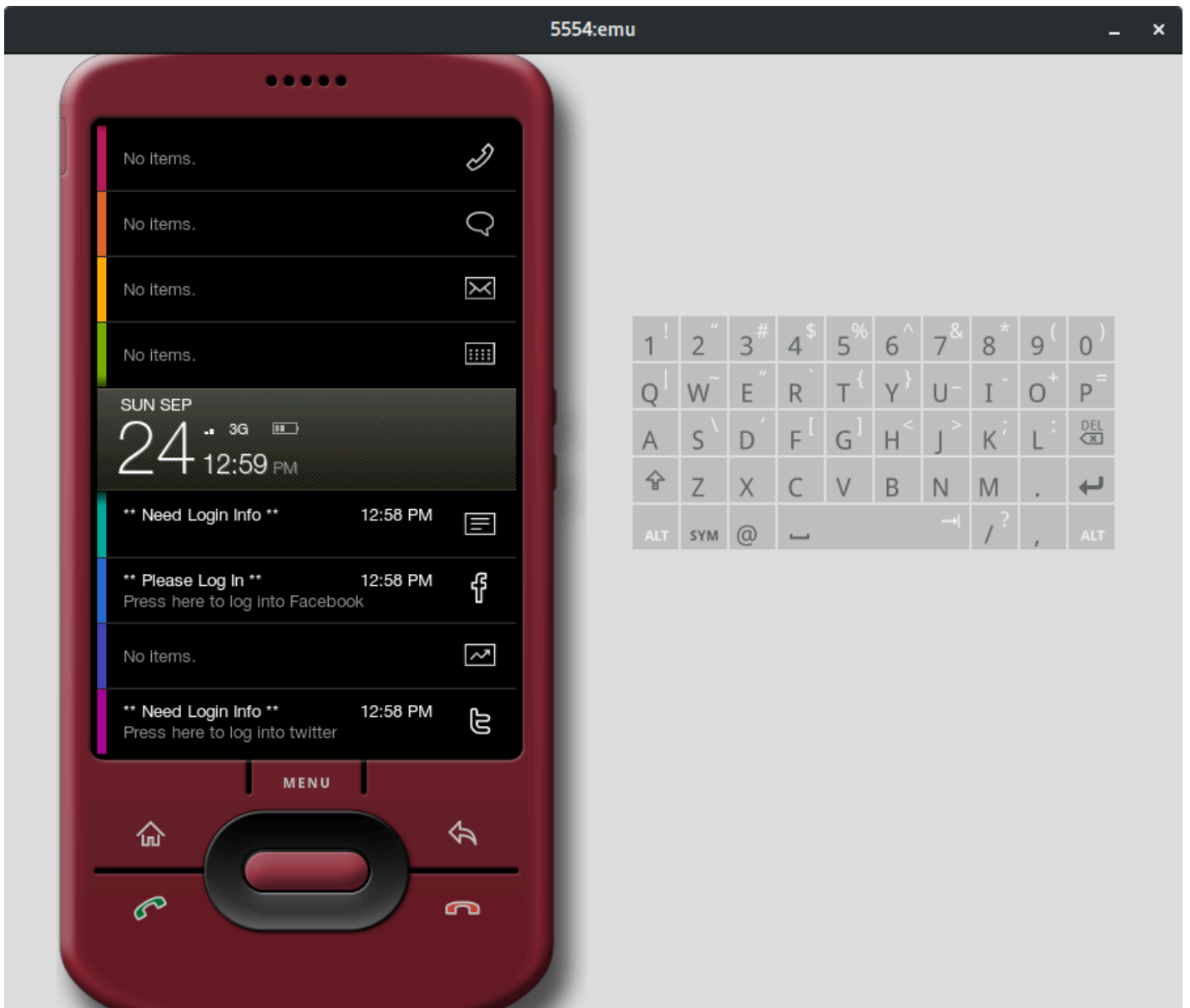
Opera

В январе 2009 года в Android Market появилась Opera Mini 4.2. Интересна она прежде всего тем, что это самая обычная Opera Mini 4.2 для телефонов, которая работает внутри эмулятора Java ME. Отображает страницы до сих пор прекрасно (спасибо серверу сжатия).



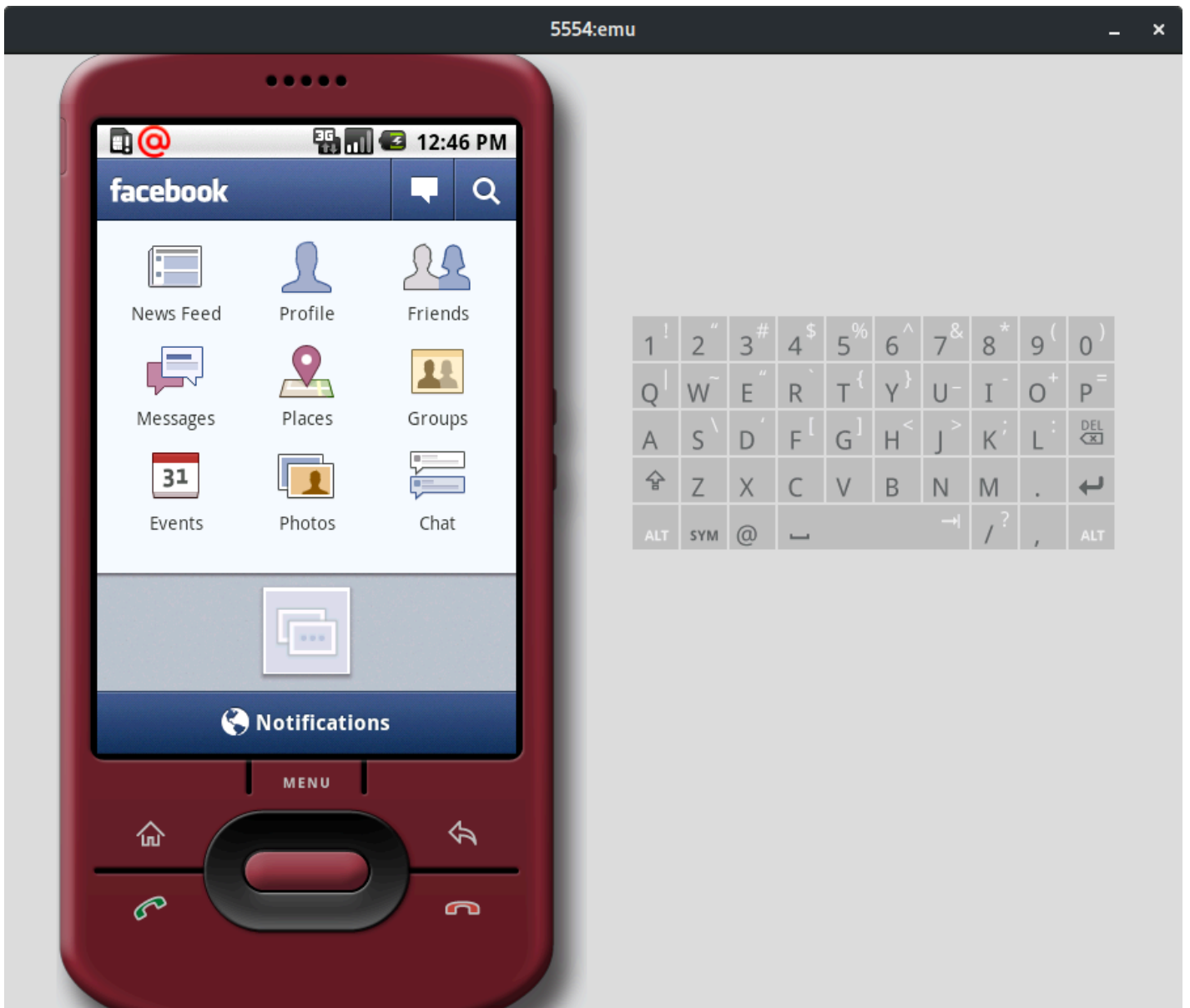
SlideScreen

Один из самых оригинальных лаунчеров для Android. Показывает много полезной информации на главном экране. Прекрасно работает в 1.5.



Facebook

В то время это была всего лишь обертка для мобильной версии сайта. До сих пор работают все функции.



ES File Explorer

Файловый менеджер с прекрасным набором функций даже по сегодняшним меркам.
Занимает всего 1.87 Мб.



Заклучение

Android 1.0 определенно очень интересная ОС. Она весьма компактна, может работать в условиях жесткого ограничения ресурсов. Архитектура Android изначально была правильной, и было бы глупо говорить о том, что в нормальную операционную систему Android превратился только недавно.

Читайте ещё больше платных статей бесплатно: <https://t.me/nopaywall>