УДК 004.4
DOI: 10.18101/978-5-9793-1397-9-61-64

ИЗ ОПЫТА НАПИСАНИЯ СКРИПТА НА SHELL В СРЕДЕ ANDROID

© **Конькова Анна Евгеньевна**
студент,
Бурятский государственный университет имени Доржи Банзарова
Россия, 670000, г. Улан-Удэ, ул. Смолина, 24а
E-mail: konkova@yandex.ru

© **Лоскова Анастасия Владиславовна**
студент,
Бурятский государственный университет имени Доржи Банзарова
Россия, 670000, г. Улан-Удэ, ул. Смолина, 24а
E-mail: konkova@yandex.ru

Возможности операционной системы семейства UNIX подталкивают пользователей к нестандартному мышлению и нахождению обходных задач для решения каких-либо задач. ОС Андроид основана на ядре Linux, которое включает в себя набор стандартных UNIX-команд и простой shell, sh. Все это значит, что мы можем не только использовать командную строку для выполнения низкоуровневых операций, но и писать shell-скрипты, которые будут выполнять функции, недоступные из графического интерфейса. В статье нами описан код в среде программирования shell для уменьшения яркости экрана смартфона при снижении заряда батареи.

Ключевые слова: shell; Android; операционная система.

Сама природа операционных систем семейства Unix делает их очень гибкими, способными на гораздо большее, чем мы привыкли думать. Это подталкивает пользователя к нестандартному мышлению и нахождению обходных путей решения каких-либо задач, созданию скриптов и программ, выполняющих обычные функции нестандартными и более эффективными способами.

С помощью командной строки многие пользователи осуществляют кастомизацию Android (изменяют вид и функциональность), получают root-права либо используют их при разработке приложения.

Практически на 99% устройств Android отсутствует командная строка, но это поправимо — доступ к командной строке можно получить[1].

Как известно, ОС Android основана на ядре Linux, которое включает в себя набор стандартных Unix-команд и простой shell, sh. Все это значит, что мы можем не только использовать командную строку для выполнения низкоуровневых операций, но и писать скрипты на shell, которые будут выполнять функции, недоступные из графического интерфейса. Остается только

[1] Русскоязычный сайт, посвященный обучению разработке приложений под Android [Электронный ресурс]. URL: http://startandroid.ru (дата обращения: 10.04.2019).

установить эмулятор терминала, и можно будет вызвать команды shell, например ping или whoami. Для задач посложнее, вроде изменения системных файлов или запуска скриптов, понадобятся root-права. В интернете можно найти способ получения root практически для любого смартфона[1].

Для экспериментов нами был выбран «рутированный» смартфон Tele2 Mini с ОС Android 5.1 Lollipop, на котором нами был установлен эмулятор терминала Termux — среди прочих он оказался намного удобнее.

Для начала мы попробовали создать скрипт, который просто звонит по конкретному номеру, по сути — ярлык на контакт:

```
!#/system/bin/sh
am start -a  android.intent.action.CALL tel:12345678901.
```

Первая строка (шебанг) указывает путь к командному интерпретатору. В Android он находится в каталоге /system/bin/sh. Следующая строка — команда активизации Call. Вместо Call можно подставить Dial, и тогда откроется окно вызова с набранным номером, но сам вызов не начнется. В теории такой скрипт должен работать, но при запуске была получена ошибка — система не распознавала команды. Как оказалось, дело было в неподходящем текстовом редакторе, в котором был написан скрипт. Этот редактор добавлял значок возврата каретки «^M» в конце строки, значок, в свою очередь, не давал правильно считать шебанг. Ошибка была исправлена в редакторе Quoda.

Когда мы снова попытались запустить скрипт, система отказала в доступе. Причина этого была непонятна, учитывая, что у скрипта были права на выполнение для всех пользователей. В терминале команда работала исправно, но только от root-пользователя. На этот раз проблема крылась в политике разрешений для приложений Android. Исправить ее удалось только установкой аддона Termux API. В дополнение к Termux API мы установили пакет команд, с помощью которых можно получить быстрый доступ к камере, уведомлениям и множеству других функций (подробнее на wiki.termux.com).

Немного освоившись с работой в shell, рассмотрим более сложный пример. В ранних версиях Android была функция, автоматически убавляющая яркость экрана при низком заряде, мы ее попробовали повторить.

Так выглядит финальный вариант скрипта:

```
while [ 1=1 ]
do
termux-battery-status>battery.txt
p='grep -o '[0-9]*[0-9]' m1 battery.txt'
if [ «$p» -lt «15»]
then
termux-brightness 70
sleep 30m
```

[1] Android [Электронный ресурс]. URL: http://ru.wikipedia.org/wiki/Android (дата обращения: 10.04.2019).

```
fi
sleep 15m
done
&.
```

Итак, сначала идет бесконечный цикл while. Termux-battery-status — это команда из пакета Termux API, которая сообщает сведения о проценте заряда и температуре батареи. Значок > означает, что мы перенаправили вывод в текстовый документ, откуда потом и считаем интересующий нас процент заряда командой grep.

Если не останавливаться на каждом параметре, то выражение «-o '[0-9]*[0-9]' m1» означает, что нам нужно только первое вхождение трех- или двузначной цифры, далее идет условный оператор. Что интересно, в shell, в отличие от того же C++, пробелы значимы. В этом мы убедились опытным путем, когда условие никак не срабатывало. Кавычки возле переменной p и числа 15 дают понять интерпретатору, что операция выполняется над числами. Это еще одна особенность shell — здесь нет привычных типов данных, есть лишь строки, которые интерпретатор воспринимает по-разному в зависимости от условия [1].

Итак, если заряд опускается ниже 15%, то яркость экрана уменьшается до 70 (не процентов, яркость задается числом от 0 до 256), после чего скрипт «засыпает» на 45 минут. В противном случае проверка происходит каждые 15 минут. Значок амперсанда означает, что скрипт будет «висеть» в системе после запуска. Еще интересный момент: после команд нет никаких знаков, но условие if закрывается словом fi, как и case словом esac. Конечно, приведенный выше вариант скрипта не самый оптимальный, однако с поставленной задачей справляется.

Но зачем все эти сложности, когда в один клик можно скачать нужное приложение с Play Market? Во-первых, скрипты очень мало весят, во-вторых, им не нужны частые обновления и, в-третьих, в них вы уж точно не обнаружите какой-нибудь вредоносный файл. Конечно, есть и минусы: при неумелом обращении с терминалом телефон может превратиться в «кирпич», поэтому необходимо знать, какую команду и для чего вы используете.

Одной из самых частых поломок смартфонов и планшетов является невозможность зарядить батарею. Многие сразу же идут в сервисный центр чтобы исправить проблему, но для тех, кто знаком с IT-технологиями, советуем попытаться решить проблему самим при помощи скриптов shell.

Литература

1. Марсикано К., Стюарт К., Филлипс Б. Программирование для профессионалов. СПб. : Питер, 2017. 688 с.

FROM THE EXPERIENCE OF WRITING A SCRIPT ON SHELL IN THE ANDROID ENVIRONMENT

Anna E. Konkova
Student,
Dorzhi Banzarov Buryat State University
24a Smolina St., Ulan-Ude 670000, Russia
E-mail: konkova@yandex.ru

Anastasia V. Loskova
Student,
Dorzhi Banzarov Buryat State University
24a Smolina St., Ulan-Ude 670000, Russia
E-mail: konkova@yandex.ru

The capabilities of the operating system of the UNIX family push users to think outside the box and find workarounds to solve any problems. Android OS is based on the Linux kernel, which includes a set of standard UNIX commands and a simple shell, sh. All this means that we can not only use the command line to perform low-level operations, but also write shell scripts that will perform functions that are not available from the graphical interface. In the article, we described the code in the Shell programming environment to reduce the brightness of the smartphone screen while reducing battery power.

Keywords: shell; Android; operating system.