

# Философия UNIX

Материал из Википедии — свободной энциклопедии

Текущая версия страницы пока не проверялась опытными участниками и может значительно отличаться от версии (https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B8%D1%8F\_UNIX&stable=1), проверенной 10 октября 2014; проверки требуют 6 правок (https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B8%D1%8F\_UNIX&oldid=66065801&diff=cur&diffonly=0).

**Философия UNIX** — это набор культурных норм и философских подходов к разработке программного обеспечения, основанных на опыте ведущих разработчиков операционной системы UNIX.

Содержание

■

1

Макилрой: Четверть века UNIX

■

2

Майк Ганцарз: Философия UNIX

■

3

Реймонд: Искусство программирования в UNIX

■

4

Цитаты

■

5

Критика

■

5.1

The UNIX-HATERS Handbook

■

6

Примечания

■

7

См. также

■

8

Ссылки

## Макилрой: Четверть века UNIX

Дуг Макилрой, изобретатель каналов UNIX и один из основателей традиции UNIX, обобщил философию следующим образом:

«Философия UNIX гласит:

Пишите программы, которые делают что-то одно и делают это хорошо.

Пишите программы, которые бы работали вместе.

Пишите программы, которые бы поддерживали текстовые потоки, поскольку это универсальный интерфейс».

Обычно эти высказывания сводятся к одному «Делайте что-то одно, но делайте это хорошо».

Из этих трёх принципов только третий является специфичным для UNIX, хотя разработчики UNIX чаще других акцентируют внимание на всех трёх принципах.

## Майк Ганцарз: Философия UNIX

В 1994 году Майк Ганцарз (англ. *Mike Gancarz*) объединил свой опыт работы в UNIX (он является членом команды по разработке системы X Window System) с высказываниями из прений, в которых он участвовал со своими приятелями программистами и людьми из других областей деятельности, так или иначе зависящих от UNIX, для создания *Философии UNIX*, которая сводится к 9 основным принципам:

1.

Красиво — небольшое.

2.

Пусть каждая программа делает что-то одно, но хорошо.

3.

Стройте прототип программы как можно раньше.

4.

Предпочитайте переносимость эффективности.

5.

Храните данные в простых текстовых файлах.

6.

Извлекайте пользу из уже существующих программных решений.

7.

Используйте языки сценариев для уменьшения трудозатрат и улучшения переносимости.

8.

Избегайте пользовательских интерфейсов, ограничивающих возможности пользователя по взаимодействию с системой.

9.

Делайте каждую программу «фильтром».

также влияло на принципы не столько вообще прикладной, а именно системной философии. Статьи в некоторых случаях являлись предметом горячих споров (монолитное ядро против микроядра):

1. *Позвольте пользователю настраивать окружение.*
2. *Делайте ядра операционной системы маленькими и легковесными.*
3. *Используйте нижний регистр и придерживайтесь кратких названий.*
4. *Не храните тексты программ в виде распечаток («Спасите деревья!»).*
5. *Не сообщайте пользователю об очевидном («Молчание — золото»).*
6. *Разбивайте сложные задачи на несколько простых, выполняемых параллельно («Мыслите „параллельно“»).*
7. *Объединённые части целого есть нечто большее, чем просто их сумма.*
8. *Ищите 90-процентное решение.*
9. *Если можно не добавлять новую функциональность, не добавляйте её («Чем хуже, тем лучше»).*
10. *Мыслите иерархически.*

## Реймонд: Искусство программирования в UNIX

Эрик С. Рэймонд (англ. *Eric S. Raymond*) в своей книге «Искусство программирования в UNIX» подытожил философию UNIX как широко используемую инженерную философию «Делай это проще, глупец» (Принцип KISS). Затем он описал, как эта обобщённая философия применима в качестве культурных норм UNIX. И это несмотря на то, что несложно найти несколько нарушений в следующей текущей философии UNIX:

- **Правило модульности:** Пишите простые части, соединяемые понятными интерфейсами.
- **Правило ясности:** Ясность лучше заумности.
- **Правило композиции:** Разрабатывайте программы так, чтобы их можно было соединить с другими программами.
- **Правило разделения:** Отделяйте правила (*policy*) от механизма (*mechanism*); отделяйте интерфейс от движка (*engine*).
- **Правило простоты:** Нацельтесь на простоту; добавляйте сложность, только где необходимо.
- **Правило экономности:** Пишите большую программу только когда другими средствами выполнить необходимую задачу не удастся.
- **Правило прозрачности:** Разрабатывайте прозрачные программы для облегчения последующего пересмотра и отладки.
- **Правило надёжности:** Надёжность — дитя прозрачности и простоты.
- **Правило представления:** Храните знания в данных так, чтобы логика программы была тупой и надёжной.
- **Правило наименьшего удивления:** При разработке интерфейса всегда делайте как можно меньше неожиданных вещей.
- **Правило тишины:** Если программе нечего сказать, пусть лучше молчит.
- **Правило восстановления:** Если надо выйти из строя, делайте это шумно и как можно быстрее.
- **Правило экономии:** Время программиста дорого; сократите его, используя машинное время.
- **Правило генерации:** Избегайте ручного набора кода; при любом удобном случае пишите программы, которые бы писали программы.
- **Правило оптимизации:** Сначала — опытный образец, потом — «причесывание». Добейтесь стабильной работы, только потом оптимизируйте.
- **Правило многообразия:** Отвергайте все утверждения об «единственно правильном пути».
- **Правило расширяемости:** Разрабатывайте для будущего. Оно наступит быстрее, чем вы думаете.

Большинство из этих норм принимается вне сообщества UNIX — даже если это было не так во времена, когда они впервые были применены в UNIX, то впоследствии это стало так. К тому же много правил не являются уникальными или оригинальными для сообщества UNIX. Тем не менее, приверженцы программирования в UNIX склоняются к тому, чтобы принять сочетание этих идей в качестве основ для стиля UNIX.

## Цитаты

- «UNIX прост. Но надо быть гением, чтобы понять его простоту» — Деннис Ритчи.
- «UNIX не предназначен для ограждения своих пользователей от глупостей, поскольку это оградило бы их и от умных вещей» — Дуг Гвин.
- «UNIX никогда не говорит „пожалуйста“» — Роб Пайк.

## Критика

### The UNIX-HATERS Handbook

Философия UNIX критиковалась в книге «The UNIX-HATERS Handbook», изданной в начале 1990-х годов.

- По мнению редакторов книги, подход UNIX приводит к появлению решений, сделанных наспех, без должного продумывания архитектуры, после чего данные решения канонизируются (enshrined), то есть объявляются вечной классикой. Например, таким решением, по их мнению, являются lock files — временные файлы без содержимого, создаваемые как пометка того факта, что какая-то программа находится в процессе исполнения.
- X Window System была подвергнута критике за отделение в ней механизма (engine) от политики (policy), что привело к отсутствию в UNIX стандарта на политики управления пользовательским интерфейсом и большим затруднениям при разработке приложений, использующих GUI.
- NFS была подвергнута критике за изначально порочный подход к архитектуре — попытку создать stateless файл-сервер при том, что это принципиально невозможно. Когда же невозможность поддержки некоторых важных вещей стала очевидной, к NFS прикрутили «костыль» под названием процесса lockd.

по, в то же время, критикуемые в этой книге подходы, начатые в UNIX, плавно обосновываются и в ОС Microsoft Windows и Apple Mac OS.

## Примечания

## См. также

- Plan 9 — разработана как наследница UNIX
- Каналы и фильтры
- Разработка программного обеспечения

## Ссылки

- Brian Kernighan, Rob Pike. The UNIX Programming Environment (<http://cm.bell-labs.com/cm/cs/upe/>) 1984
- Rob Pike. Notes on Programming in C (<http://www.lysator.liu.se/c/pikestyle.html>) 11 сентября 1989
- Richard Gabriel. The Rise of Worse is Better ([http://xahlee.org/UnixResource\\_dir/\\_fastfood\\_dir/worse-is-better.html](http://xahlee.org/UnixResource_dir/_fastfood_dir/worse-is-better.html)) // Lisp: Good News, Bad News, How to Win Big (<http://www.ai.mit.edu/docs/articles/good-news/good-news.html>) 1991
- Peter H. Salus. A Quarter Century of UNIX. Addison-Wesley, 31 мая 1994. ISBN 0-201-54777-5
- Eric S. Raymond. Philosophy (<http://www.faqs.org/docs/artu/philosophychapter.html>) // The Art of UNIX Programming (<http://www.catb.org/~esr/writings/taoup>). Addison-Wesley, 17 сентября 2003. ISBN 0-13-142901-9
- M. D. Schroeder, D. D. Clark, J. H. Saltzer, D. H. Wells. Final Report of the Multics Kernel Design Project (<http://citeseer.ist.psu.edu/schroeder77final.html>) 1977
- Joel Spolsky. Biculturalism (<http://www.joelonsoftware.com/articles/Biculturalism.html>) (англ.). *Joel on Software* (14 декабря 2003). — Взгляд Windows-разработчика на различия двух культур. Архивировано из первоисточника 5 февраля 2012 (<http://www.webcitation.org/65E5Vn2JZ>).
- Денис Смирнов. Классический UNIX-way или «компьютер для профессионала» ([http://freemsource.info/wiki/Stat'ja\\_Klassicheskijj\\_Unix\\_Way](http://freemsource.info/wiki/Stat'ja_Klassicheskijj_Unix_Way)) (рус.) (2004). — Разъяснение базовых принципов Unix простым языком. Архивировано из первоисточника 5 февраля 2012 (<http://www.webcitation.org/65E5WIDqT>).

Источник — «[https://ru.wikipedia.org/w/index.php?title=Философия\\_UNIX&oldid=74728461](https://ru.wikipedia.org/w/index.php?title=Философия_UNIX&oldid=74728461)»

- 
- Последнее изменение этой страницы: 20:21, 24 ноября 2015.
  - Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.