



Лаборатория системного программирования и
информационных технологий СПбГУ

Традиционные методы разработки для ОС Linux

Сергей Леви

- ▶ GNU
- ▶ GNU Compiler Collection (gcc)
- ▶ GNU make
- ▶ GNU autotools
- ▶ GNU debugger
- ▶ Открытые IDE
- ▶ Открытые инструменты профилировки и отладки



Linux – developer's heaven

- ▶ Большое число проверенных временем инструментов разработки.
- ▶ Гибкость и модульность (в широком смысле этого слова)
- ▶ Простота
- ▶ Множество доступных библиотек
- ▶ Open source и всё, что это значит

Проект GNU

- ▶ Начат Ричардом Столлманом в 1983 г.
- ▶ GNU = GNU's Not UNIX
- ▶ Цель
 - ▶ *“a sufficient body of free software [...] to get along without any software that is not free.”*
 - ▶ Свободная ОС
- ▶ GCC, glibc, coreutils, binutils, bash, GNOME ...



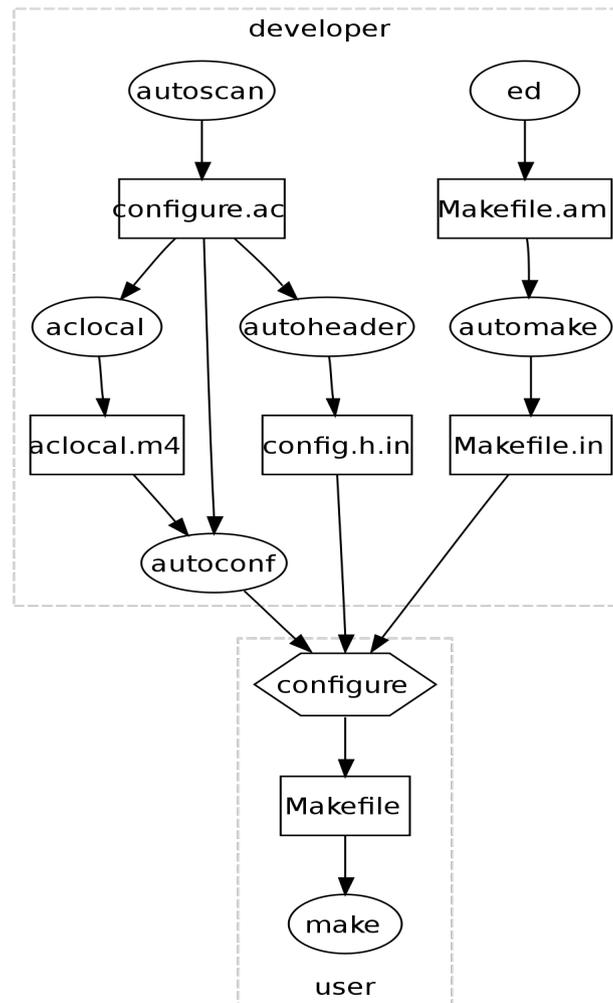
- ▶ Первый релиз в 1987 году (в виде GNU C compiler)
- ▶ Текущая версия – 4.5.1
- ▶ Поддерживаемые языки:
C, C++, Java, Ada, Objective C, Objective C++, Fortran и другие
- ▶ Целевые архитектуры
IA32(x86), x86-64, ARM, PowerPC, ARM, MIPS, SPARC, Amtel AVR и много, много других



Компиляция – это просто!

- ▶ Компиляция (а также ассемблирование и линковка) кода на C
 - ▶ `gcc mysource.c`
- ▶ Компиляция кода на C++
 - ▶ `g++ myprogram.cpp`
- ▶ Компиляция приложения, состоящего из нескольких модулей
 - ▶ `gcc -o myprogram file1.c file2.c file3.c`
- ▶ Компиляция и линковка со внешней библиотекой
 - ▶ `gcc -lm myprogram.c`

- ▶ make осуществляет сборку
- ▶ configure (автосгенерированный скрипт) создаёт Makefile из Makefile.in
- ▶ autoconf генерирует скрипт configure из configure.ac
- ▶ automake генерирует Makefile.in на основании Makefile.am и configure.ac



Система сборки GNU (2/2)

- ▶ Makefile – непосредственные указания для сборки проекта
- ▶ Makefile.in – по сути, почти Makefile, но без привязки к конкретной платформе
- ▶ configure – shell-скрипт, собирающий данные о системе, на которой производится сборка и на основе Makefile.in создающий Makefile
- ▶ Makefile.am – данные о структуре исходного кода
- ▶ configure.ac – данные о конфигурации сборки

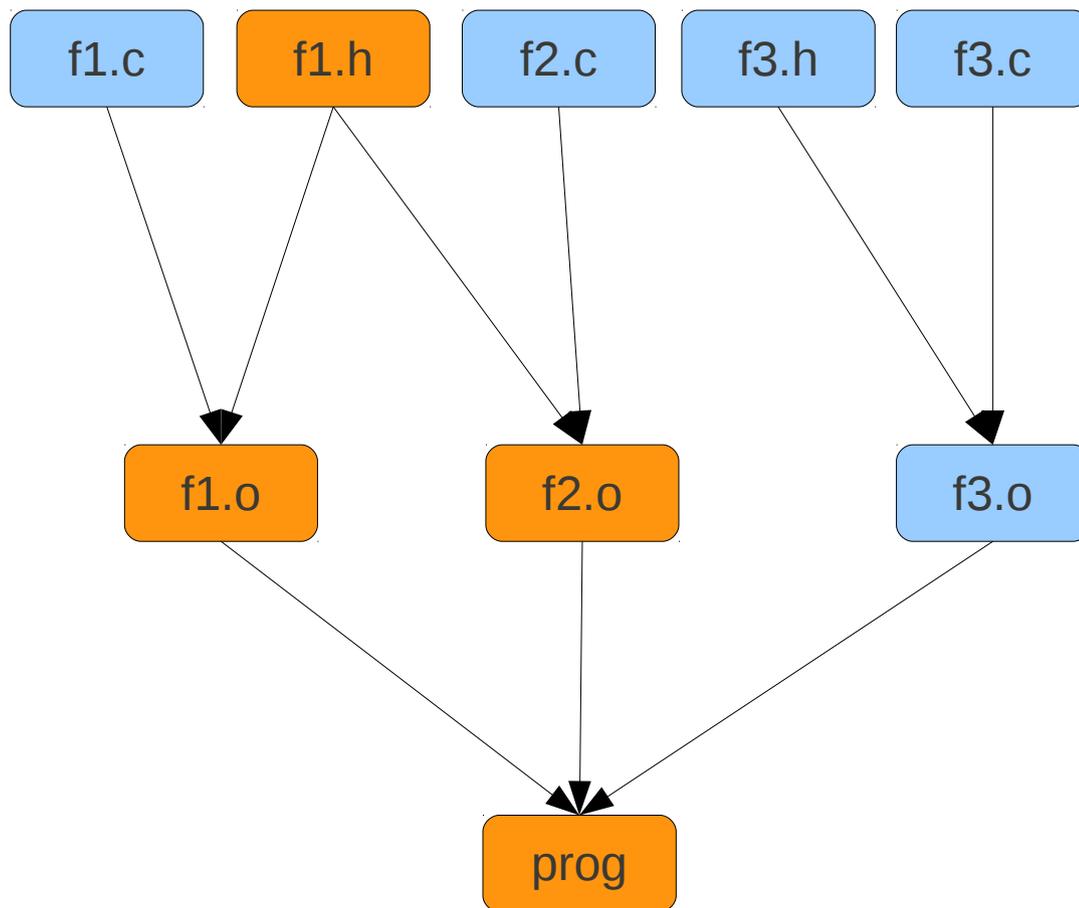


(GNU) make

- ▶ Переписана с UNIX-программы 1977 г., созданной в Bell Labs.
- ▶ Входит в стандарт POSIX
- ▶ Задача:
 - ▶ упростить сборку больших проектов
 - ▶ отслеживать зависимости между модулями
 - ▶ компилировать только обновлённые части проекта

(GNU) make – принцип действия

- ▶ Определяются цели (targets) – файлы
- ▶ Для целей определяются реквизиты (файлы / другие цели) и/или команды
- ▶ Если реквизит обновлён позднее цели – он пересобирается



(GNU) make – пример

```
all: prog
f1.o: f1.c f1.h
    gcc -c -o $@ $<
f2.o: f2.c f1.h
    gcc -c -o $@ $<
f3.o: f3.c f3.h
    gcc -c -o $@ $<
```

```
prog: f1.o f2.o f3.o
    gcc -o prog $^

clean:
    rm -rf *.o prog

.PHONY: all clean
```

Используем систему управления пакетами apt (в Debian-подобных дистрибутивах)

- ▶ Скачиваем пакеты исходного кода
 - ▶ `apt-get source <название пакета>`
- ▶ Устанавливаем пакеты, от которых зависит данный пакет:
 - ▶ `apt-get build-dep <название пакета>`
- ▶ Запускаем скрипт конфигурации `configure`
- ▶ Запускаем сборку: `make`



Альтернативные системы автоматизации сборки

- ▶ Qmake (о нём подробнее в следующих лекциях)
- ▶ CMake
- ▶ SCons
- ▶ ...

GNU debugger (gdb)

- ▶ Первая версия выпущена в 1986 г. Ричардом Столлманом.
- ▶ Поддерживает отладку C, C++, D, Objective C, Fortran, Pascal, Modula-2, Ada
- ▶ Интерфейс командной строки
- ▶ Существует множество графических оболочек
- ▶ Поддерживает удалённую отладку
- ▶ Имеет back-end для отладки ядра Linux



```
user@host:~$ gdb hello
GNU gdb (GDB) 7.1-ubuntu
This GDB was configured as "i486-linux-gnu".
Reading symbols from
/home/user/hello...done.
(gdb) break print
print_with_timestamp printf
printf@plt
(gdb) break print_with_timestamp
Breakpoint 1 at 0x80484ff: file hello_util.c,
line 15.
(gdb) run
Starting program: /home/user/hello

Breakpoint 1, print_with_timestamp
(msg=0x8048650 "Hello, world!") at
hello_util.c:15
15 {
```

```
(gdb) l
10 #include <stdio.h>
11
12 #include "hello_util.h"
13
14 int print_with_timestamp(char *msg)
15 {
16     char buf [TIME_STR_LEN];
17     time_t rawtime;
18     struct tm *timeinfo;
19
(gdb) p msg
$1 = 0x8048650 "Hello, world!"
(gdb) c
Continuing.
Sun Oct 10 15:41:51 2010: "Hello, world!"
Program exited normally.
(gdb) quit
```

Прощай, командная строка

- ▶ Свободные IDE для разработки на C\C++
 - ▶ Qt Creator (о нём в следующий раз)
 - ▶ Eclipse
 - ▶ NetBeans
 - ▶ Anjuta
 - ▶ Kdevelop
 - ▶ ...



- ▶ GNU profiler (gprof) – профилировщик. Требует специальной сборки.
- ▶ Valgrind – отладка утечек памяти, профилировка, построение дерева вызовов
- ▶ KCacheGrind – графический анализатор вывода Valgrind
- ▶ strace – трассировка системных вызовов
- ▶ OProfile – профилировщик, использующий счётчики производительности процессора
- ▶ perfmon – профилировщик, требует перекомпиляции ядра
- ▶ PTLsim – симулятор процессора x86



Для дополнительного чтения

- 1) Wikipedia
- 2) Документация по GCC (<http://gcc.gnu.org/onlinedocs/>)
- 3) Документация по GNU make
(<http://www.gnu.org/software/make/manual/make.html>)
- 4) Autobook AKA “The Goat Book” (<http://sources.redhat.com/autobook/>)
- 5) Документация по automake
(<http://sources.redhat.com/automake/automake.html>)
- 6) Документация по autoconf
(<http://www.gnu.org/software/autoconf/manual/autoconf.html>)
- 7) Документация по GDB
(<http://www.gnu.org/software/gdb/documentation/>)
- 8) “The Linux Development Platform” R.U. Rehman, C. Paul
(<http://www.informit.com/content/downloads/perens/0130091154.pdf>)

Вопросы?

sergeyle@gmail.com

