

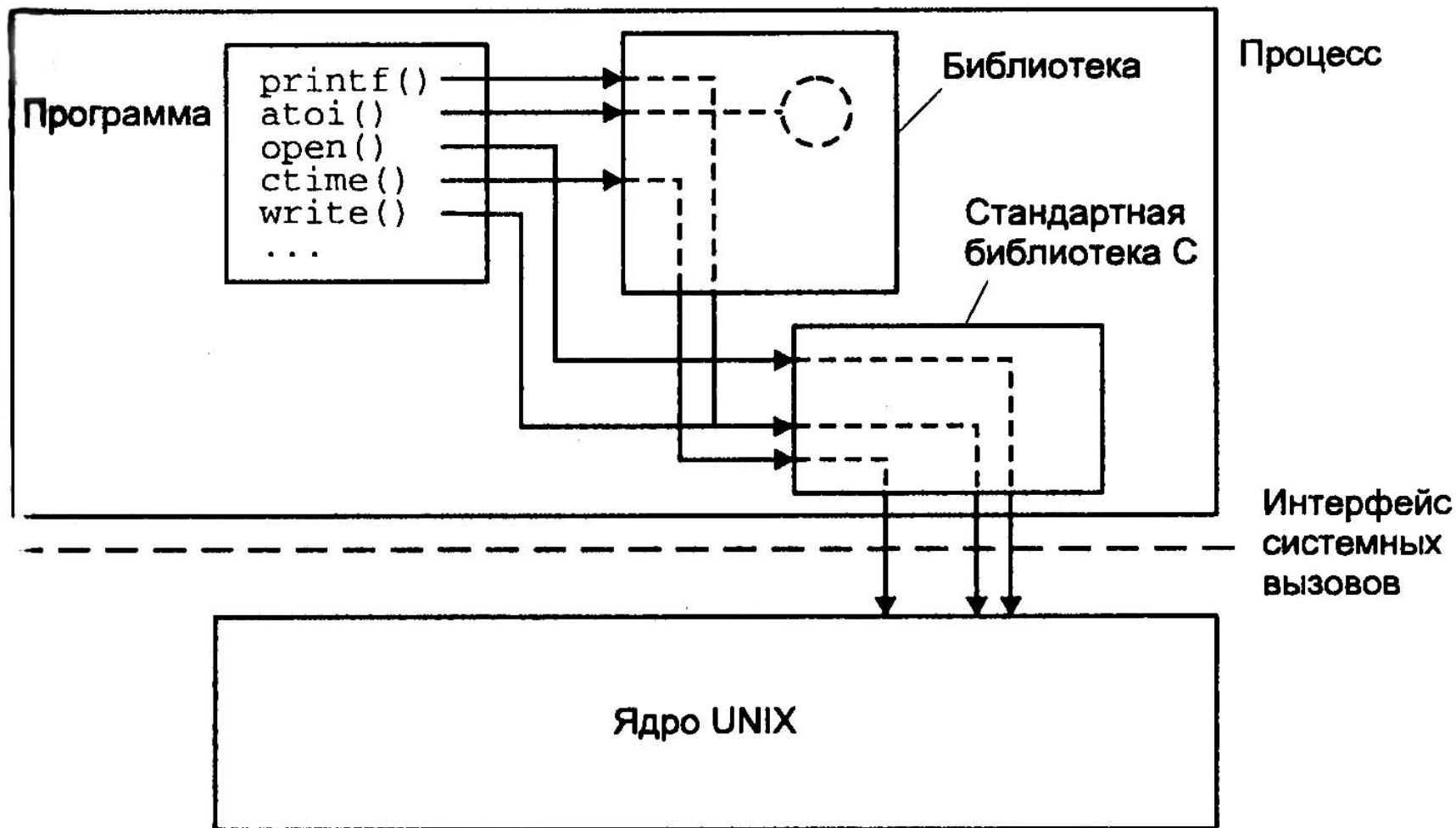
Среда программирования UNIX

- | Среда программирования UNIX.
- | Разработка ПО в UNIX, CVS.

GCC – GNU Compiler Collection или GNU C Compiler

- И Разработан Ричардом Столлманом в 1985, поддерживается FSF
- И Поддерживаемые языки: Ada (GCC для Ada aka GNAT), C, C++ (GCC для C++ aka G++), Fortran (GCC для Fortran aka GFortran), Java (GCC для Java aka GCJ), Objective-C, Objective-C++. Частично: Pascal, D, Modula-2, Modula-3, Mercury, VHDL и PL/I.
- И Поддерживаемые платформы: Alpha, ARM, Atmel AVR, Blackfin, H8/300, System/370, System/390, IA-32 (x86 и x86-64), IA-64 в том числе и «Itanium», Motorola 68000/88000, MIPS, MSP430 Texas Instruments, PA-RISC, PDP-11, PowerPC, SuperH, HC12, SPARC, VAX, семейство Renesas R8C/M16C/M32C, семейство MorphoSys
- И Средство отладки – gdb плюс многочисленные интерфейсы

Системные вызовы и библиотечные функции



Системные вызовы (man 2 имя_вызова), библиотечные функции (man 3 имя_функции)

Этапы создания программы

- текстовый редактор: `vi limits.c getlimit.c`
 - ▮ `limits.c getlimit.c` Исходные файлы
- компиляция (cc): `gcc -c getlimit.c limits.c`
 - ▮ `limits.o getlimit.o` Объектные файлы
- компоновка (ld): `gcc getlimit.o limits.o -o limits`
 - ▮ `limits` Исполняемый файл/Библиотечка
- исполнение: `./limits`
- отладка: `gdb, xgdb(ключ -g, наличие *.c)`
 - ▮ `gcc -g getlimit.c limits.c; gcc -g getlimit.o limits.o -o limits`
 - ▮ `gdb limits`

Makefile

I make [-f Makefile]

```
limits: limits.o getlimit.o
    gcc limits.o getlimit.o -o limits
limits.o: limits.c
    gcc -c limits.c
getlimit.o: getlimit.c
    gcc -c getlimit.c
```

I С макросами вида строка1=строка2

```
CC=gcc
CFLAGS=-g
#
limits: limits.o getlimit.o
    $(CC) $(CFLAGS) limits.o getlimit.o -o limits
limits.o getlimit.o: limits.c getlimit.c
```

Распространение пакетов, переносимость, Autocconf (2.6.1)

- ┌ Autocconf – это программа для генерирования шелл-скрипта “configure” автоматически конфигурирующего пакет ПО в исходных текстах для обеспечения его сборки в UNIX–подобных системах.
- ┌ Для работы программы autocconf требуется программа m4 (GNU вариант).
- ┌ Поскольку генерируется шелл-скрипт, при сборке ПО сама программа autocconf может быть и не установлена на целевой системе.
- ┌ Для создания скрипта configure, нужно написать входной файл configure.in и запустить autocconf .
- ┌ Программа ifnames, входящая в состав пакета, помогает при написании или модификации configure.in, осуществляя поиск директив #if, #elif, #ifdef, #ifndef в исходных текстах ПО

Autosconf: подготовка программного пакета к распространению

```
Файлы исходных текстов -> [autoscan] -> [configure.scan] -> configure.in

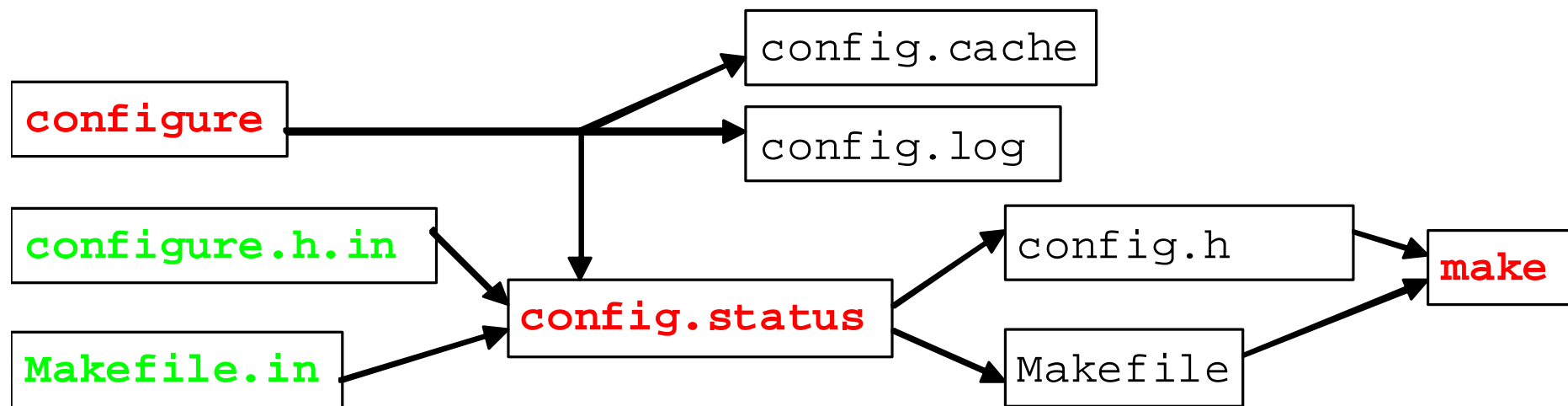
configure.in --.      .-----> autoconf -----> configure
                +----+
[aclocal.m4] --+      `--.
[acsite.m4] ---'      |
                    +--> [autoheader] -> [config.h.in]
[acconfig.h] ----.   |
                +-----'
[config.h.top] ---+
[config.h.bot] --'

Makefile.in -----> Makefile.in
```

Обычный порядок вызова макросов Autosconf:

1. Проверка программ
2. Проверка библиотек
3. Проверка заголовочных файлов
4. Проверка определений типов
5. Проверка структур
6. Проверка характеристик компилятора
7. Проверка библиотечных функций
8. Проверка системных сервисов

Конфигурирование пакета ПО, подготовленного Autoconf



Совместная разработка ПО, CVS

- | CVS – Concurrent Versions System поддерживает историю дерева каталогов с исходным кодом. Является развитием RCS (Revision Control System)
- | CVSweb – WWW интерфейс к CVS
- | Есть Windows порт CVS, под названием CVSNT

- | Дальнейшее развитие CVS – Subversion (SVN): KDE, Gnome, Samba, модуль Apache, GCC, Mono...
- | Git – еще одна распространенная система контроля версий: ядро Linux, Xorg, GNU core utilities...

- | CVS использует переменную окружения CVSROOT для определения местоположения репозитория
- | `CVSROOT=/usr/src/myproject; export CVSROOT`

Последовательность действий при работе с CVS

Если необходимо существующий проект разместить в CVS, обычно выполняют импорт исходников из тек. каталога: `cvс -m "imported src" limits limits start`

1. `$ cvs checkout myprog` - извлечение рабочего каталога из репозитория
2. `$ vi limits/limits.c` - редактирование исходного кода
3. `$ cvs update` - объединение изменений
4. `$ cvs commit limits.c` - фиксация изменений

`$ cvs log main123.c` - отслеживание изменений

`$ cvs diff -u -r 1.1 -r 1.3 main123.c` - просмотр изменений

Конфликт и его разрешение в CVS

Конфликт:

```
$ cvs update - объединение изменений
```

...

```
rcsmerge: warning: conflicts during merge
```

```
cvs update: conflicts found in main123.c
```

...

Разрешение:

```
$ vi main123.c
```

```
$ cvs commit main123.c - фиксация изменений
```

SVN

- | `svnadmin create /usr/src/limits`
- | `cd ~/src/orig`
- | `svn import limits file:///usr/src/limits -m "imported src"`
- | `svn list file:///usr/src/limits`
- | `cd ~/src; svn checkout file:///usr/src/limits`
- | `cd limits; vi limits.c; svn update;`
- | `svn status; svn diff; svn commit`
 - | В случае конфликта: `svn revert limits.c` или `svn resolved limits.c; svn commit`
- | `svn log limits.c -v; svn list -v`