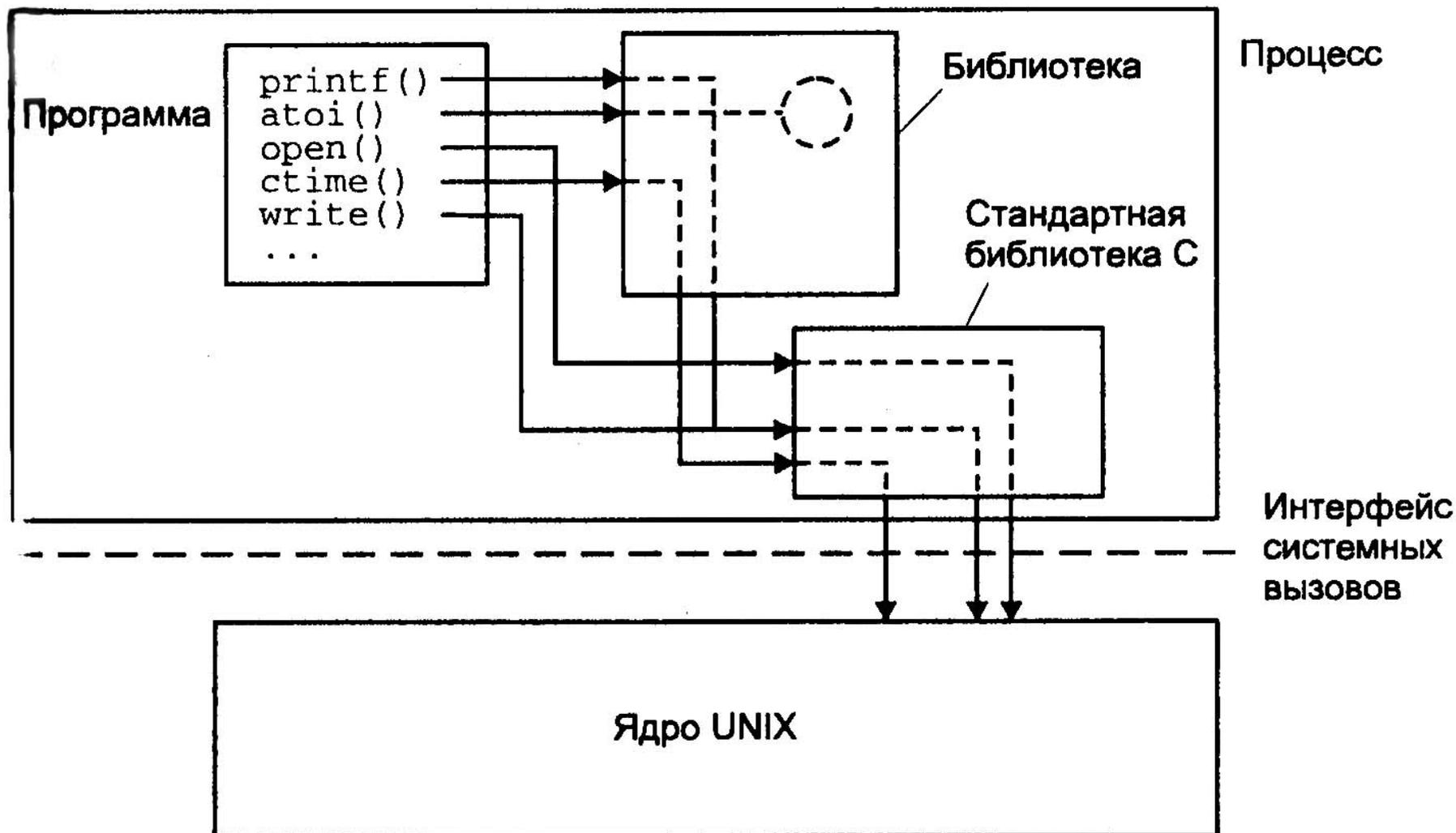


Среда программирования

- | Системные вызовы (map 2 имя_вызова) и библиотечные функции (map 3 имя_функции) (см. следующий слайд Системные вызовы и библиотечные функции)
- | Создание программ
 - | Исходные тексты
 - | Этапы разработки
 - | Makefile и configure
- | Форматы исполняемых файлов
- | Исполнение программ в ОС Unix

Системные вызовы и библиотечные функции



темы «Программный интерфейс»

- | Фазы разработки приложения
- | Основные утилиты, языки
- | Организация разработки, управление версиями
- | Компиляция, компоновка, исполнение
- | Библиотеки: использование, создание
- | Отладка, профилирование, оптимизация
- | Автоматизация разработки ПО, make
- | Распространение пакетов, переносимость, autoconf
- | Совместная разработка ПО, CVS

Этапы создания программы

- | текстовый редактор
- | файл.с Исходные файлы
 - | компиляция (cc): cc -c a.c; cc -c b.c
- | файл.o Объектные файлы
 - | компоновка (ld): cc a.o b.o -o prog
- | prog Исполняемый файл/Библи.
- | Исполнение: ./prog
- | Отладка: gdb, xgdb (prog, *.c)

Makefile

I make [-f Makefile]

```
| prog: a.o b.o  
|     cc a.o b.o -o prog  
| a.o: incl.h a.c  
|     cc -c a.c  
| b.o: incl.h b.c  
|     cc -c b.c
```

I С макросами вида строка1=строка2

```
| CC=gcc  
| CFLAGS=-g  
| #  
| prog: a.o b.o  
|     $(CC) $(CFLAGS) a.o b.o -o prog  
| a.o b.o: incl.h
```

Распространение пакетов, переносимость, Autocconf

- 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

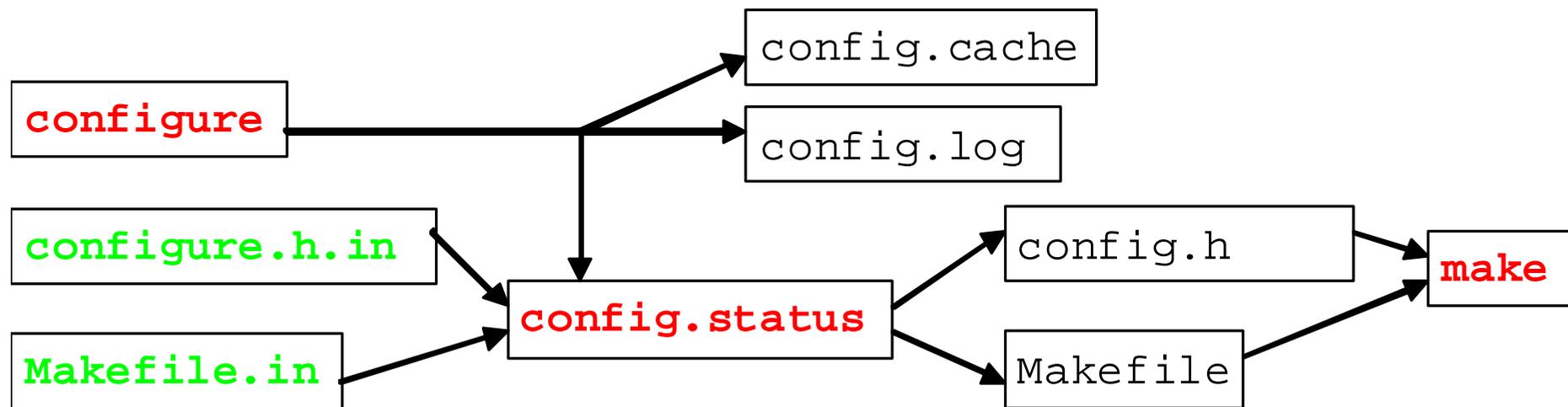
configure.in --.      .-----> autoconf -----> configure
                +----+
[aclocal.m4] --+      `--.
[acsite.m4] ---'      |
                  +--> [autoheader] -> [config.h.in]
[acconfig.h] ----.   |
                +-----'
[config.h.top] ---+
[config.h.bot] --'

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 – Control Versions System поддерживает историю дерева каталогов с исходным кодом
- CVS использует переменную окружения CVSROOT для определения местоположения репозитария
- `CVSROOT=/usr/src/myproject; export CVSROOT`

Последовательность действий при работе с CVS

1. `$ cvs checkout myprog` - извлечение рабочего каталога из репозитория
2. `$ vi myprog/main123.c` - редактирование исходного кода
3. `$ cvs update` - объединение изменений
4. `$ cvs commit main123.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
```