

Разработка кроссплатформенных
программных систем.
Лекция №0. Введение

П.Н. Советов

РТУ МИРЭА, 2022

0. Введение.

1. Разработка переносимых приложений на языке C.

2. Встраивание и расширение скриптовых языков.

3. Создание простых скриптовых языков.

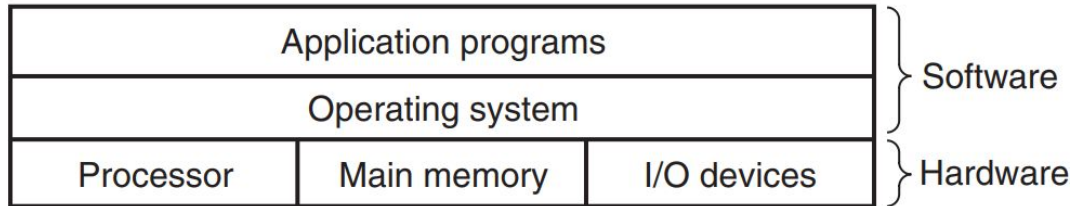
4. Техники интерпретации для обеспечения переносимости.

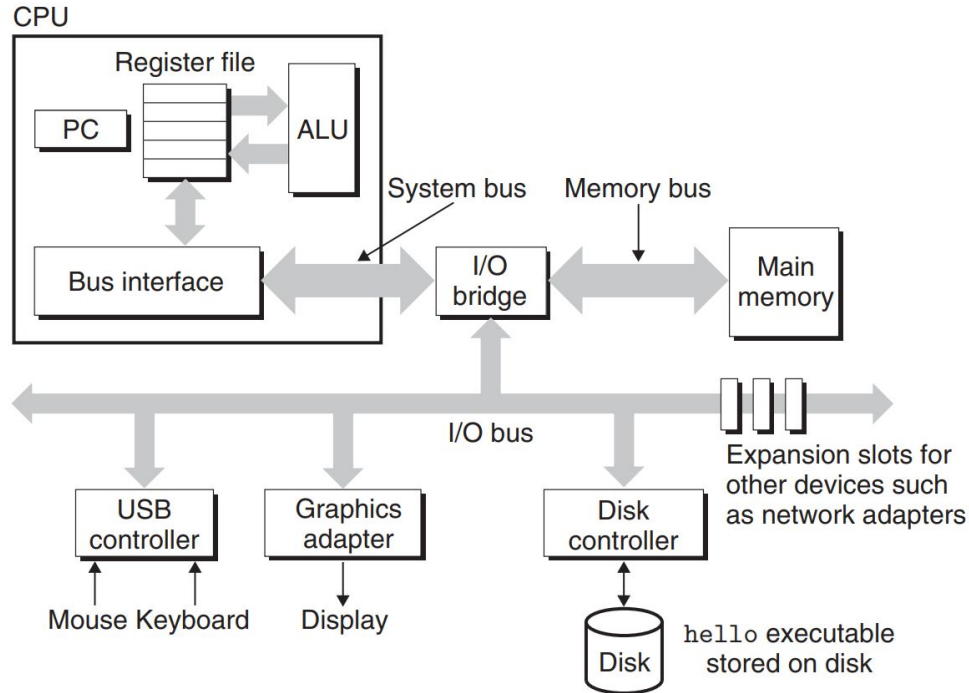
5. Техники компиляции для обеспечения переносимости.

6. Кроссплатформенные графические интерфейсы.

Платформа – окружение, в котором выполняется ПО (примеры: настольная, мобильная, веб, для встраиваемых систем).

Различают **программную** и **аппаратную** платформы.



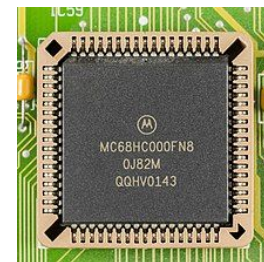




- ARC,
- ARM,
- ARM64,
- IA-64,
- LoongArch,
- m86k,
- MIPS,
- Nios II,
- OpenRISC,
- PA-RISC,
- PowerPC,
- RISC-V,
- s390,
- SuperH,
- SPARC,
- x86,
- Xtensa.



RISC-V: The Free and Open RISC
Instruction Set Architecture





Программная платформа – набор системного ПО, обеспечивающего выполнение прикладного ПО.



Системное ПО – программные средства для обеспечения функционирования и разработки ПО.

Примеры системного ПО: ОС, драйверы, библиотеки, ассемблеры, компиляторы, эмуляторы аппаратных средств.



macOS



Перенос или портирование ПО (porting software) – адаптация ПО для новой платформы.

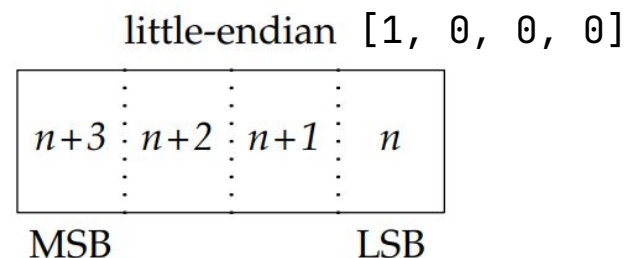
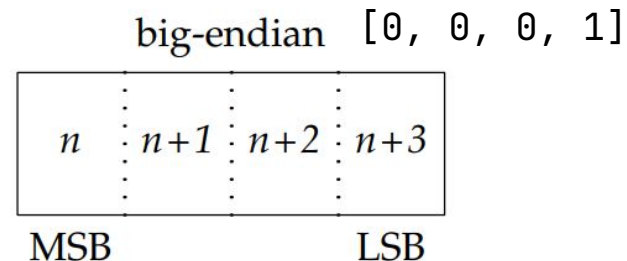
Переносимое ПО (portable software) – ПО, перенос которого на новую платформу стоит значительно дешевле, чем повторная разработка.

Кроссплатформенное ПО (cross-platform software) – ПО, предназначенное для работы на нескольких, заранее заданных платформах.

- Несовместимые наборы команд.
- Различный объем памяти.
- Различный порядок байт при хранении чисел.
- Различные варианты выравнивания данных.
- Различные возможности периферийных устройств.
- ...

- От старшего к младшему (big-endian).
- От младшего к старшему (little-endian).
- Настраиваемый.

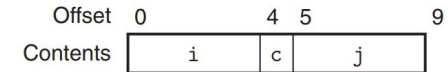
Operating system	Processor architecture	Byte order
FreeBSD 8.0	Intel Pentium	little-endian
Linux 3.2.0	Intel Core i5	little-endian
Mac OS X 10.6.8	Intel Core 2 Duo	little-endian
Solaris 10	Sun SPARC	big-endian



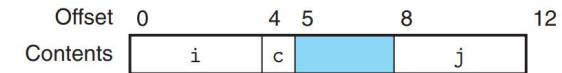
Выравнивание для типа данных размером N означает, что эти данные должны располагаться по адресу, кратному N .

Нарушение этого ограничения может привести к потере быстродействия и, в некоторых архитектурах, к ошибке выполнения программы (для SSE, к примеру, требуется выравнивание на границу 16 байт).

arch	Align:	char	short	int	long	ptr	long-long	u8	u16	u32	u64
i386		1	2	4	4	4	4	1	2	4	4
i686		1	2	4	4	4	4	1	2	4	4
alpha		1	2	4	8	8	8	1	2	4	8
armv4l		1	2	4	4	4	4	1	2	4	4
ia64		1	2	4	8	8	8	1	2	4	8
mips		1	2	4	4	4	8	1	2	4	8
ppc		1	2	4	4	4	8	1	2	4	8
sparc		1	2	4	4	4	8	1	2	4	8
sparc64		1	2	4	4	4	8	1	2	4	8
x86_64		1	2	4	8	8	8	1	2	4	8



Адрес j выравнен:



API (application programming interface) – программный интерфейс приложения.

API – способ использования на уровне исходных текстов функциональности программы или библиотеки со стороны другого ПО.

ABI (application binary interface) – двоичный интерфейс приложения.

ABI – низкоуровневый, аппаратно-зависимый способ взаимодействия ПО в виде машинного кода (выбор регистров, порядок размещения данных в стеке, ...).

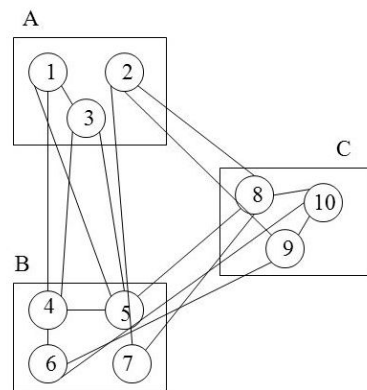
- Несоблюдение стандартов при использовании языка программирования, библиотек и инструментального ПО.
- Использование непереносимых библиотек и особенностей ABI.
- Применение низкоуровневых трюков в коде.
- ...

Правила проектирования переносимого ПО в целом мало отличаются от общих правил проектирования хорошего ПО.

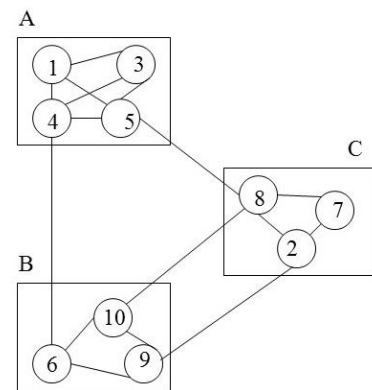
Сопряжение (coupling) – степень взаимозависимости модулей в программе.

Связность (cohesion) – степень, в которой элементы одного модуля связаны друг с другом.

Необходимы **слабое сопряжение** и **сильная связность**.



Bad modularization:
low cohesion, high coupling



Good modularization:
high cohesion, low coupling

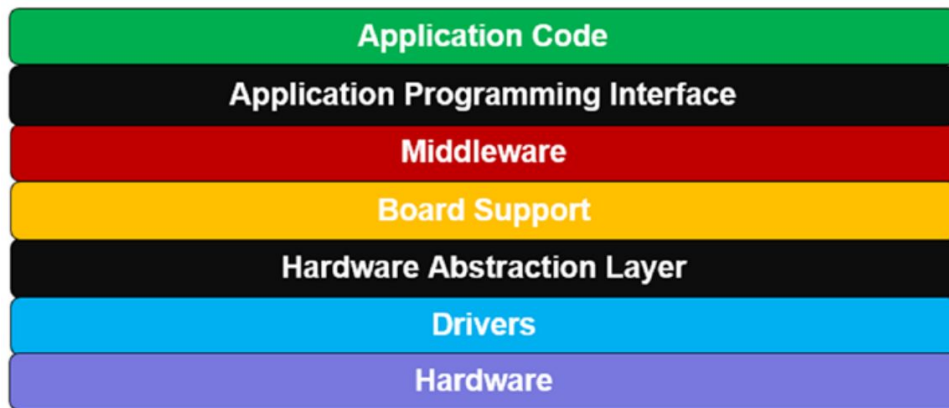
Принцип единственной ответственности (single-responsibility principle) — каждый модуль или функция должны заниматься одним-единственным делом.

Абстрагирование — выделение сущностных характеристик объекта и избавление от низкоуровневых деталей.

Инкапсуляция — выделение в модуле открытого интерфейса и закрытой реализации.

Для обеспечения переносимости важным является **выделение платформо-зависимого кода в отдельные модули** с переносимым интерфейсом.

Слой аппаратных абстракций (hardware abstraction layer) – низкоуровневый API для переносимого доступа к аппаратно-зависимым функциям.



1. **Объединение** — разработка программ, которые работают на каждой из выбранных платформ благодаря тому, что в программу включен весь необходимый код, для поддержки каждой из платформ.

Недостатки: слишком большие объемы кода, нередко слишком высокая его сложность, трудность доработки и тестирования программ.

2. **Пересечение** — написание как можно большей части кода в обобщенном виде, чтобы программа работала без изменений на любой из платформ.

Недостатки: потенциальная потеря быстродействия и даже отдельных функциональных возможностей.

Графика игры The Great Escape (1986) в версиях C64 и IBM PC выглядит так же, как и в версии для ZX Spectrum, который обладал самыми слабыми граф. возможностями.



Требования к кроссплатформенному ПО:

- максимальное использование **функций** каждой из платформ (пример: где-то есть сенсорный экран, а где-то – мышь).
- высокое **быстродействие** (запуск с помощью виртуализации может этого не обеспечить).
- высокая степень **взаимодействия** с каждой платформой (пример: обмен данными через буфер обмена).

