

# LitePAC – инструментарий разработчика проектов автоматки

Михаил Ильин

[ilyin\\_mikhail@inbox.ru](mailto:ilyin_mikhail@inbox.ru)

# АСУТП

---

- уровень контролера управления процессом;
- промышленные сети – контролеры, датчики, исполнительные механизмы;
- образ процесса – срез состояний входов, выходов аналоговых, дискретных плат контролеров;
- прикладные протоколы передачи данных.

# ПЛК и ПАК

---

ПЛК (программируемые логические контролеры) – специализированные устройства для автоматизации технологических процессов.

- появились в 60-х годах;
- пришли на смену релейных схем;
- перезаписываемая программа исполнения;
- специализированные среды разработки.

# ПЛК и ПАК

---

ПАК (программируемые автоматизированные контролеры)

- ОС общего применения (Linux, WinCE);
- кросс-компиляторы GNU с поддержкой C/C++, библиотеки для работы с периферией;
- приобретение коммерческих среды разработки и исполнения;
- разработка собственного набора программ.

# LitePAC

---

lite [lait] прил.; прост.; = light легкий, простой.

- свободный инструментарий разработчика и среда исполнения;
- простота добавления протокольных модулей и аппаратных платформ;
- использование идиом и традиций принятых в мире Linux/Unix;
- привлечение сообщества свободного ПО к данной проблематике.

# Набор утилит LitePAC

---

- программы для разработки;
- программа среды исполнения и менеджер образов процесса;
- аппаратные и протокольные модули ввода/вывода.

# Программы для разработки

---

- компилятор (*vmlang*);
- ассемблер (*vmasm*);
- линкер (*vmlink*);
- утилита сборки (*vmc*);
- прикладные библиотеки.

# Компилятор (*vmlang*)

---

Значительно упрощенный вариант языка ANSI C с дополнительными синтаксическими конструкциями для работы с входными и выходными образами.

Исходный текст  $\longrightarrow$  Ассемблерный листинг

```
$ vmlang -o file.vS file.vm
```

# Компилятор (*vmlang*)

---

- лексический анализатор;
- синтаксический анализатор (нисходящий парсер с предсказанием);
- библиотека для создание узлов abstract syntax tree;
- библиотека для формирование intermediate representation (IR);
- реерhole оптимизация;
- генерация target кода.

# Компилятор (*vmlang*)

---

В языке поддерживаются следующие типы данных:

- *int* целое число, 32 бита;
- *float* число с плавающей точкой одинарной точности, 32 бита;
- *char* печатный символ, 8 бит;
- *string* ASCII строка;

# Компилятор (*vmlang*)

---

- привычные конструкции условий и циклов;
- массивы перечисленных типов данных;
- аргументы ф-ий передаются по значению и по ссылкам;
- аргументы ф-ий могут иметь значения по умолчанию.

# Компилятор (*vmlang*)

---

- нет указателей;
- нет составных объектов типа структур.

# Компилятор (*vmlang*)

---

Язык имеет конструкции, позволяющие адресовать входные/выходные образы модулей ввода/вывода.

Поддерживаемые области ввода/вывода:

- область входов ( $I$ );
- область выходов ( $Q$ ).

# Компилятор (*vmlang*)

---

Типы данных адресуемых в областях ввода/вывода:

- байт  $B$ , 1 байт;
- короткое слово  $S$ , 2 байта;
- слово  $L$ , 4 байта;
- число с плавающей точкой одинарной точности  $F$ , 4 байта.

# Компилятор (*vmlang*)

---

Адрес ввода/вывода формируется следующим образом:

```
$<ID области><смещение><тип> [ ' [ 'индекс' ] ' ] ] [ .B<бит> ]
```

Пример:

```
a = $I35S.B10;
```

```
for (i = 0; i < n; i++)  
    $Q14B[i] = 0xed;
```

# Ассемблер (*vmasm*), линкер (*vmlink*)

---

Ассемблер:

Ассемблерный листинг → Объектный файл

```
$ vmasm -o file.vo file.vS
```

Линкер:

Объектные файлы → Исполняемый файл

```
$ vmlink -e init -o file.v file1.vo file2.vo
```

# Утилита сборки (*vtc*).

---

Стандартный конвейер сборки:

*cc* → *as* → *ld*

Сборка LitePAC:

*vmlang* → *vmasm* → *vmlink*

Последовательность вызова утилит обернута утилитой *vtc*.

# Прикладные библиотеки

---

Библиотеки расширяют функционал отсутствующий в самом языке.

- битовые операции;
- работа со строками;
- преобразование типов;
- и т.д.

# Пример

---

```
#include <bits.h>

void sample()
{
    int num, i;
    /* Читаем слово из области входов. */
    num = $I3L;
    /* Сдвиг на 4 бита вправо. */
    num = rshift(num, 4);

    if (num != 0) {
        for (i = 0; i < 16; i++)
            /* Запись в область выходов. */
            $Q0B[i] = 0xff;
    }
}
```

# Сборка и запуск

---

```
$ vmc -e sample -I vminclude/ -o sample.v  
sample.vm vmlib/bits.vo
```

```
$ vm-lib-test -e sample -p vmlibcall/  
sample.v
```

# Среда исполнения и менеджер образов

---

Среда исполнения (*vmachined*):

- стековая виртуальная машина (*libvmrt*);
- работает с несколькими типами задач;
- простой FIFO планировщик;
- поддержка прерываний с приоритетом;
- синхронизирует входы/выходы с менеджером образов (*hal-mgr*).

# Среда исполнения и менеджер образов

---

Типы задач:

- background задача;
- циклическая задача;
- задачи запускаемые по прерываниям (32 уровня приоритета).

# Среда исполнения и менеджер образов

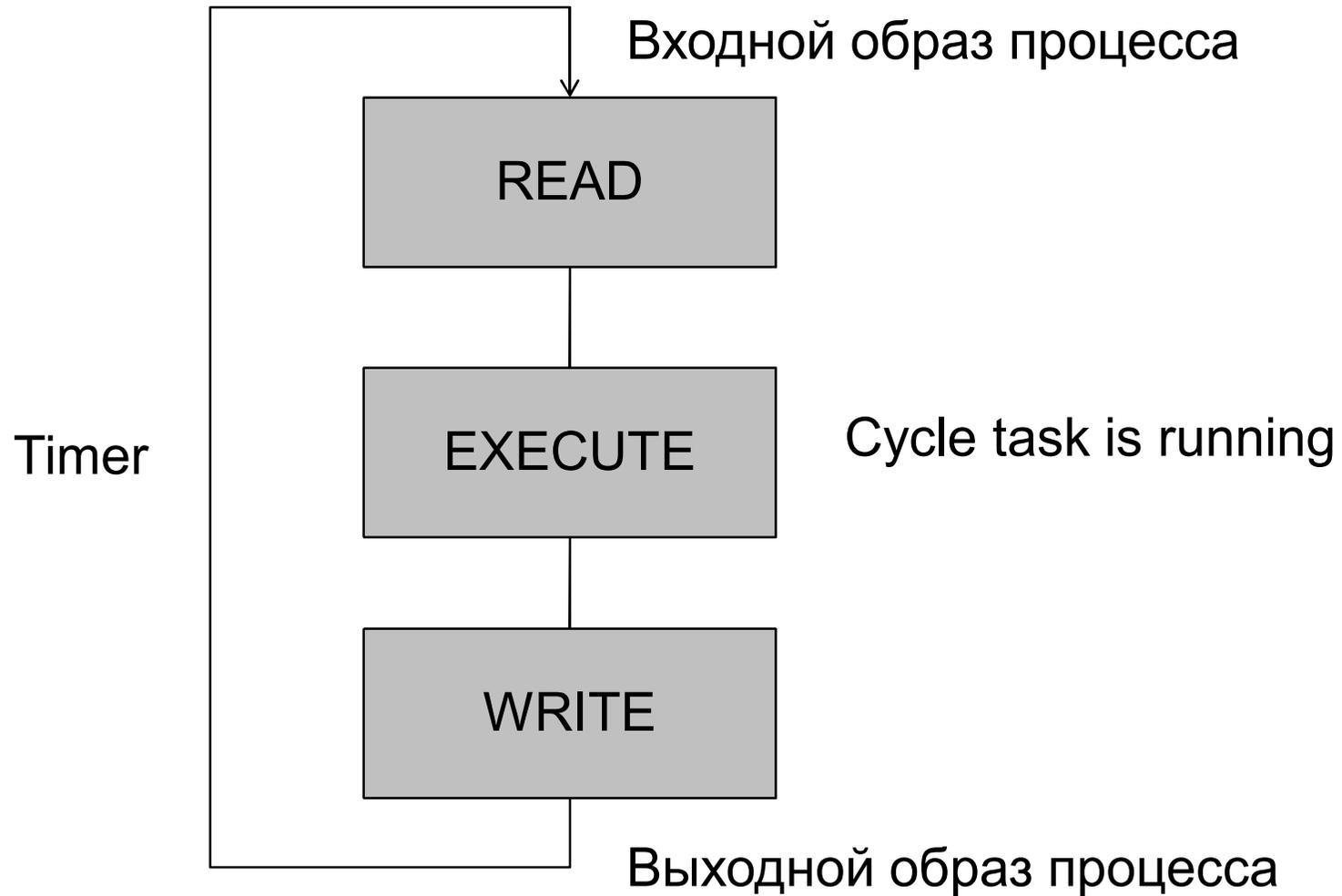
---

Циклическая задача:

- запускается по таймеру;
- инициирует синхронизацию образа входов с *hal-mgr* до запуска самой задачи;
- результатом выполнения является обновление образа выходов;
- инициирует обратную синхронизацию выходов *hal-mgr* по завершению задачи

# Среда исполнения и менеджер образов

---



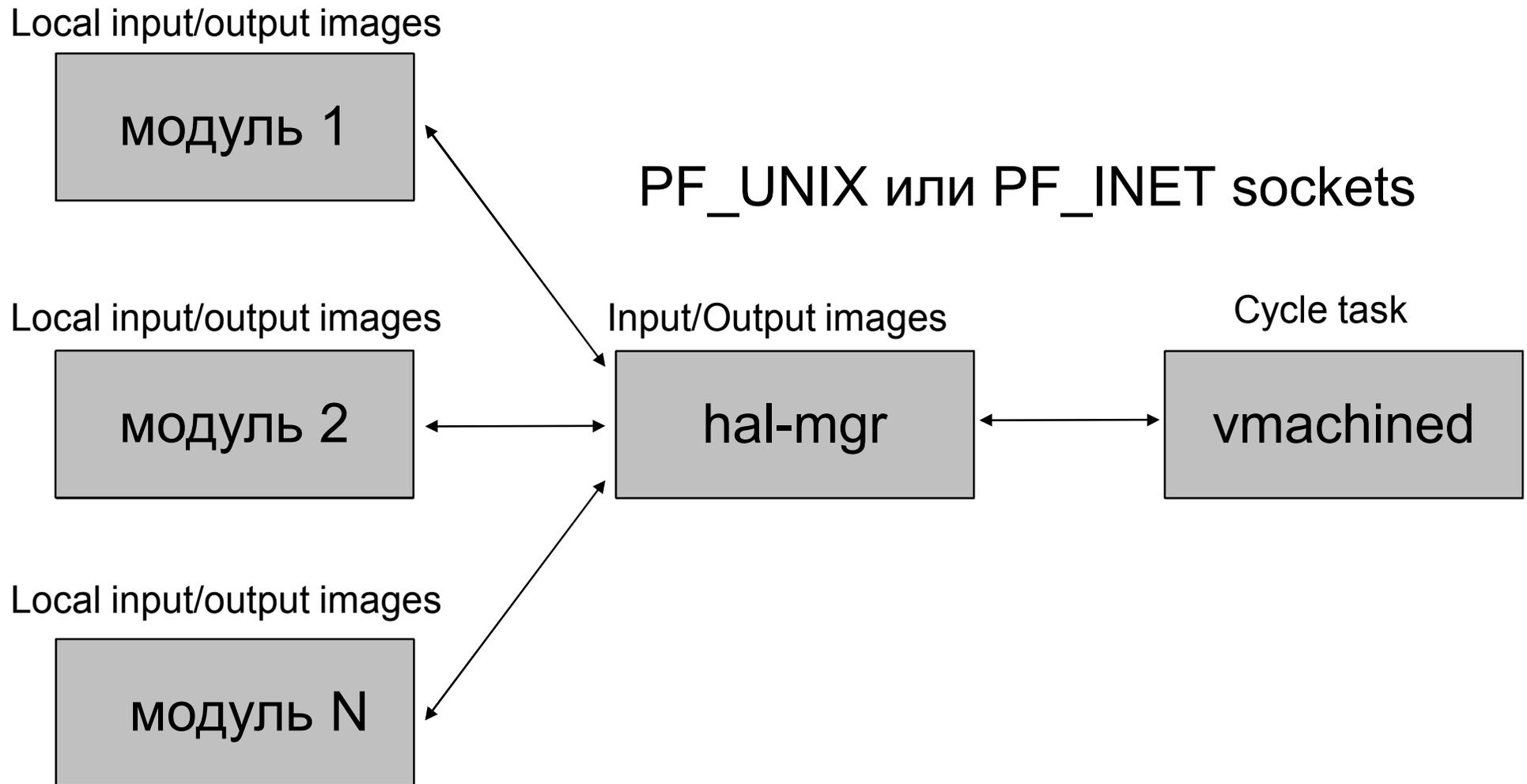
# Среда исполнения и менеджер образов

---

Задача менеджера образов (*hal-mgr*) – читать и писать локальные входные/выходные образы аппаратных и протокольных модулей.

# Среда исполнения и менеджер образов

---



# Среда исполнения и менеджер образов

---

Протокол hal (*libhal*), **текстовый** протокол:

- сигнализация о начале и конце цикла;
- синхронизация образов между средой исполнения и менеджером образов;
- синхронизация локальных образов модулей с конечными образами менеджера.

# Аппаратные и протокольные модули

---

Модули работают с *hal-mgr* в рамках протокола *hal*. Синхронизируют свои локальные образы входов и выходов по команде от *hal-mgr*.

# Аппаратные и протокольные модули

---

## Модули LitePAC:

- *lincond* – модуль для работы с контролерами ICPDAS Lincon;
- *modbusd* – реализация коммуникационного протокола Modbus TCP.

**Спасибо за внимание**

[www.litepac.org](http://www.litepac.org)