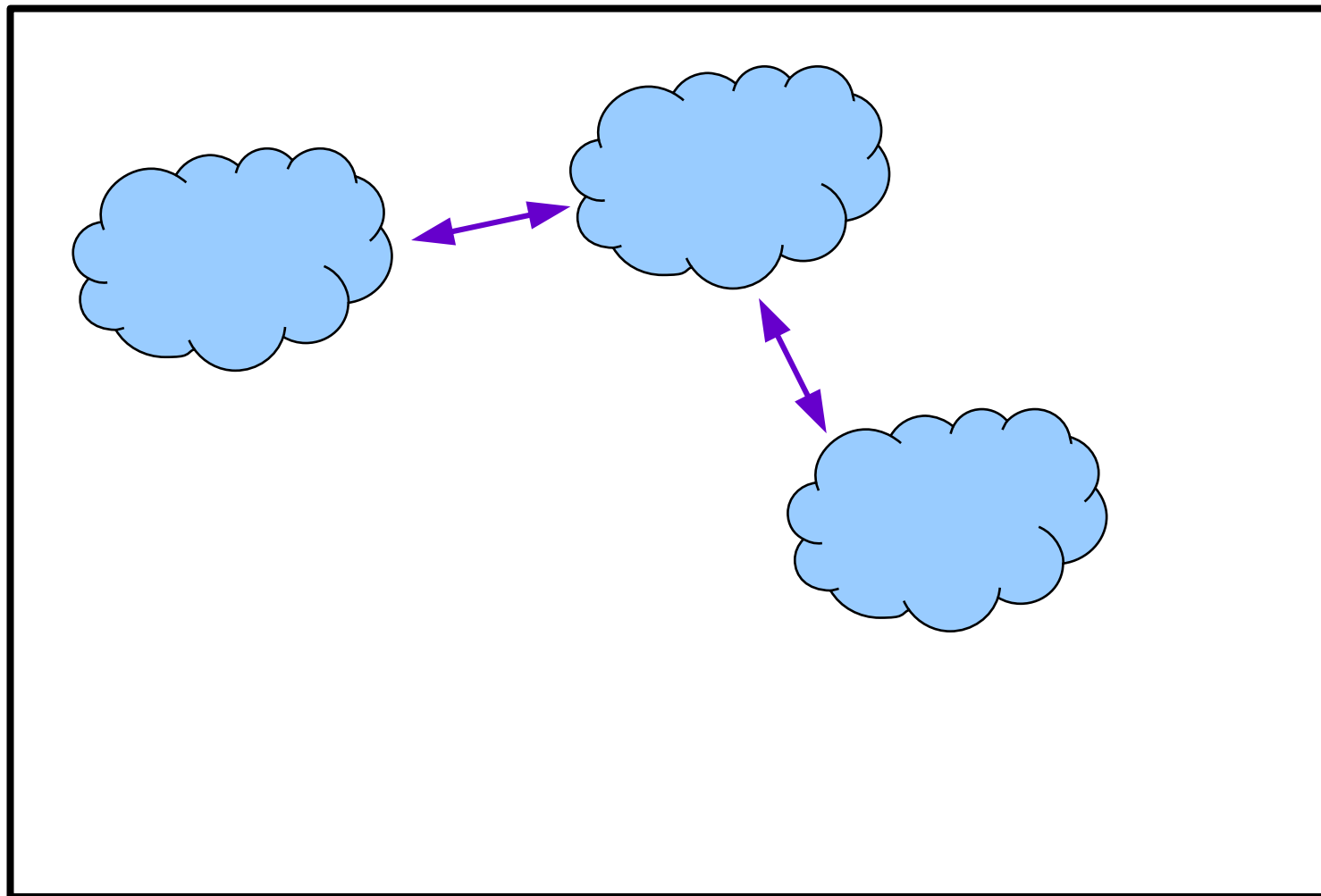
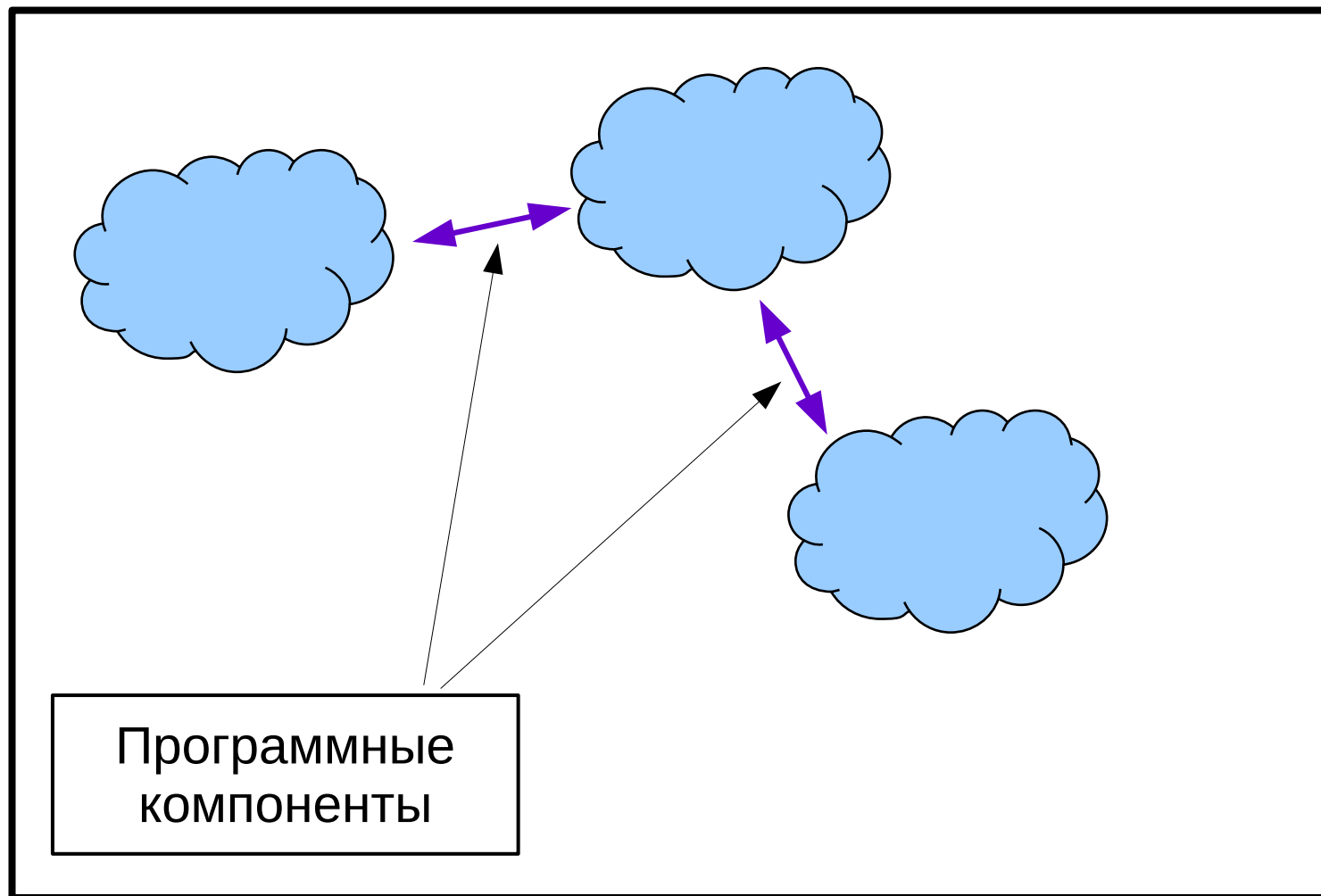
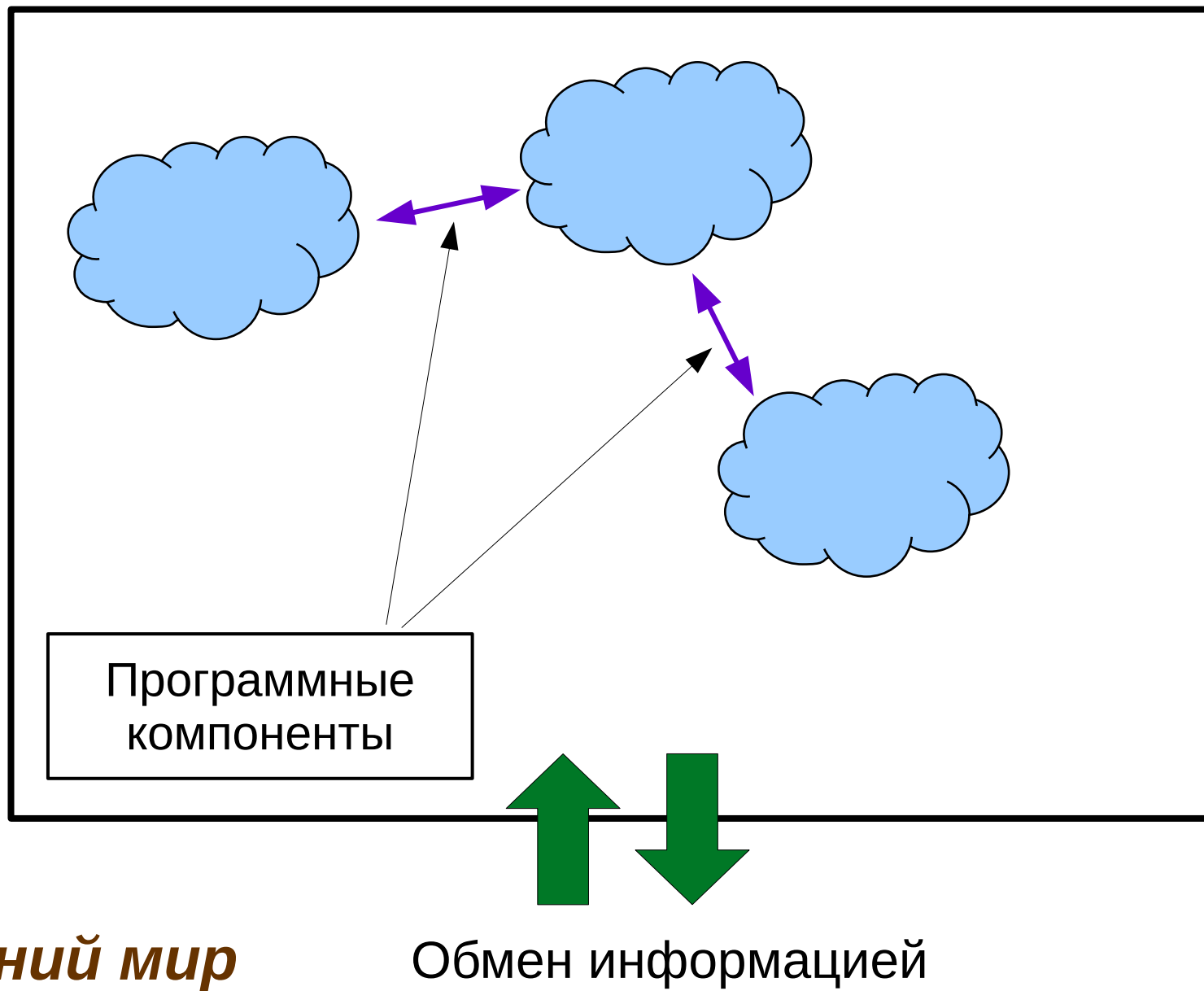


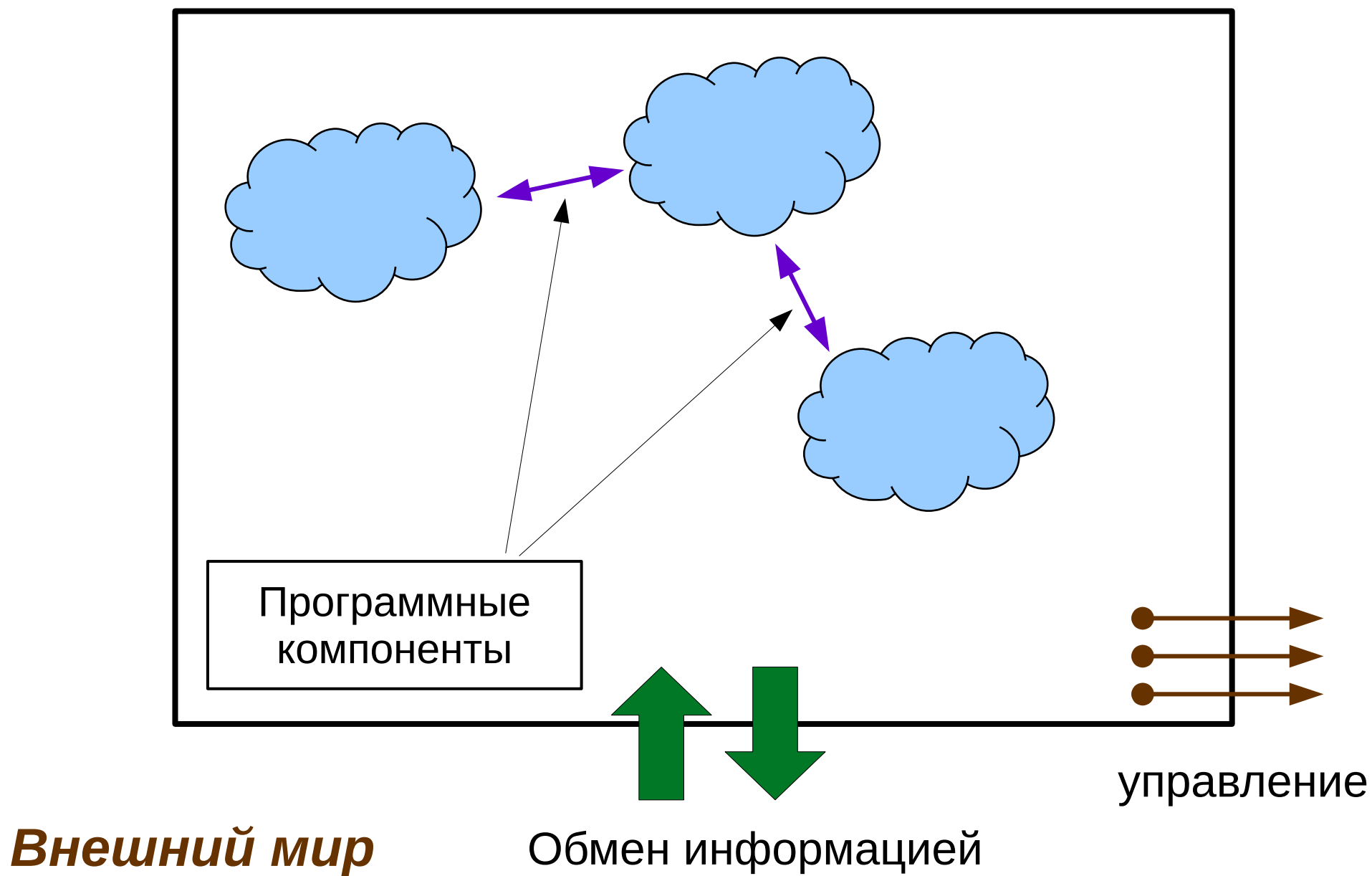


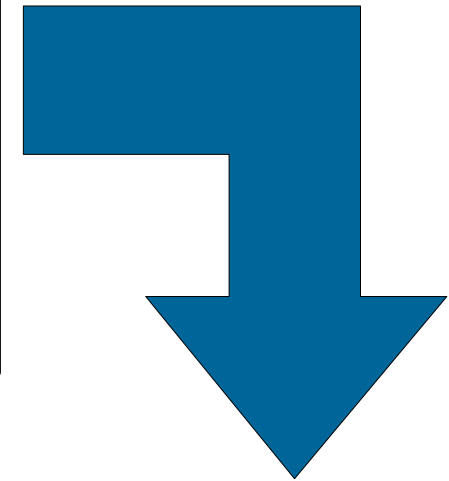
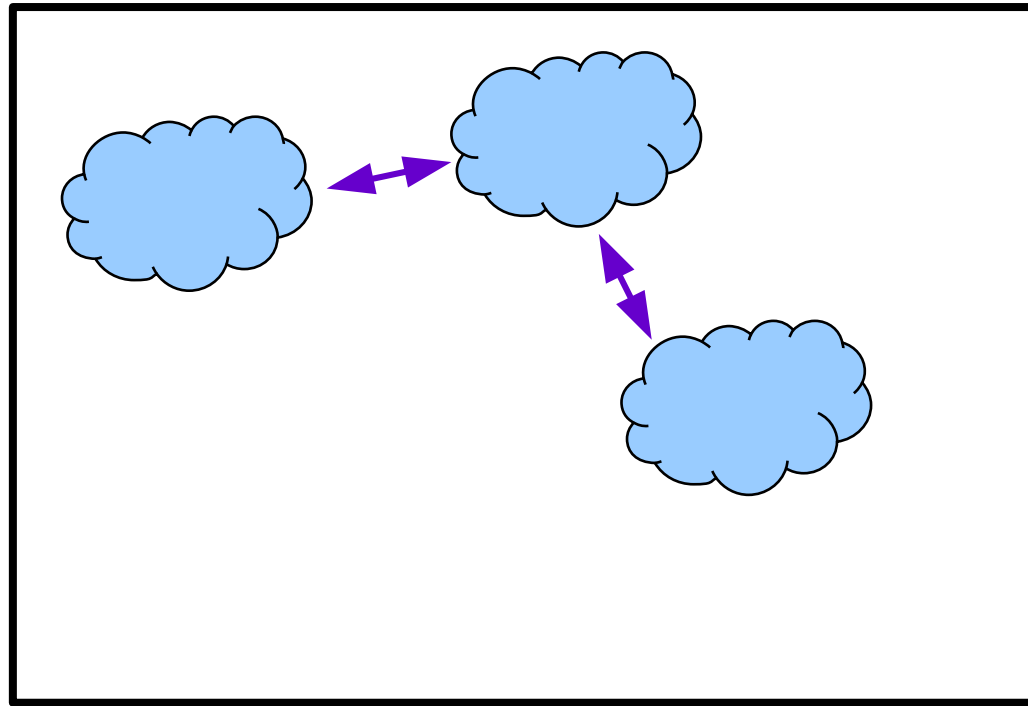
Промежуточные компоненты



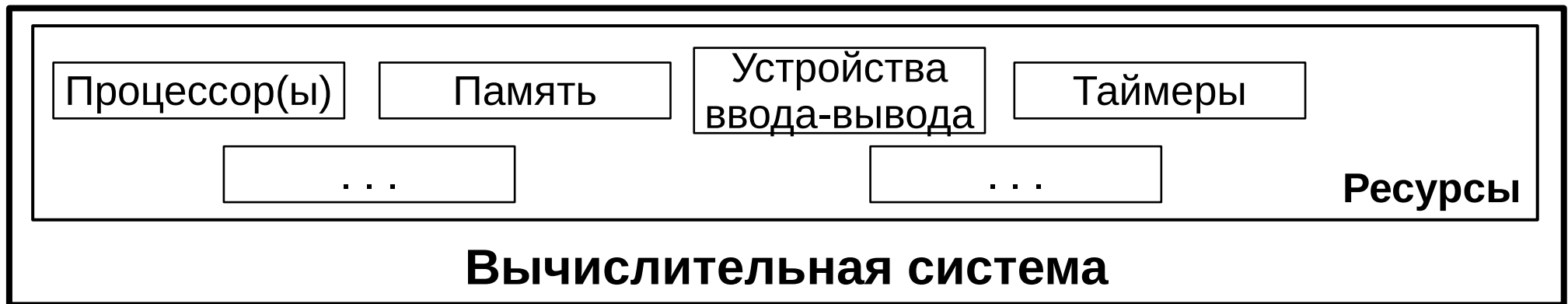
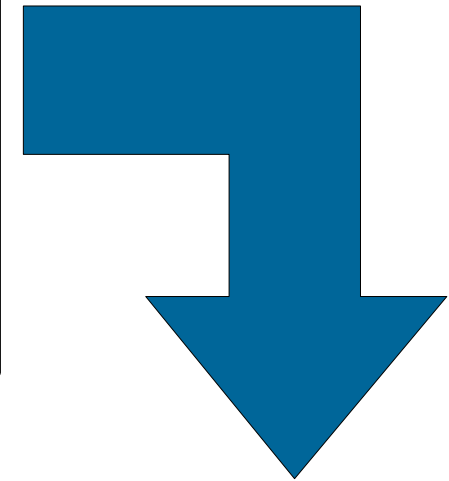
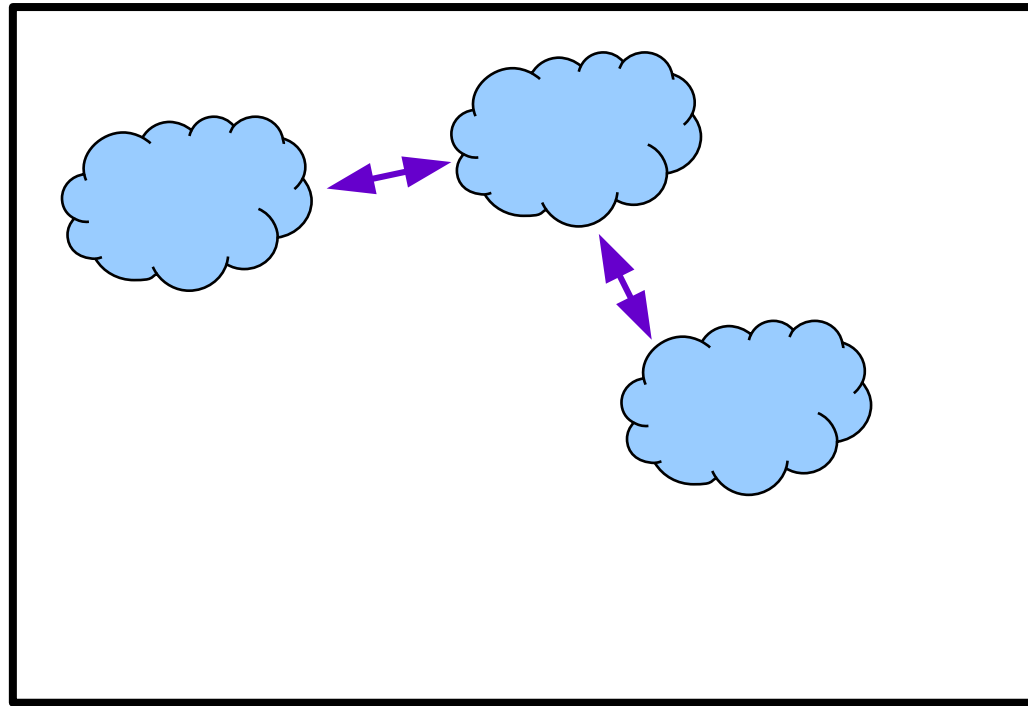


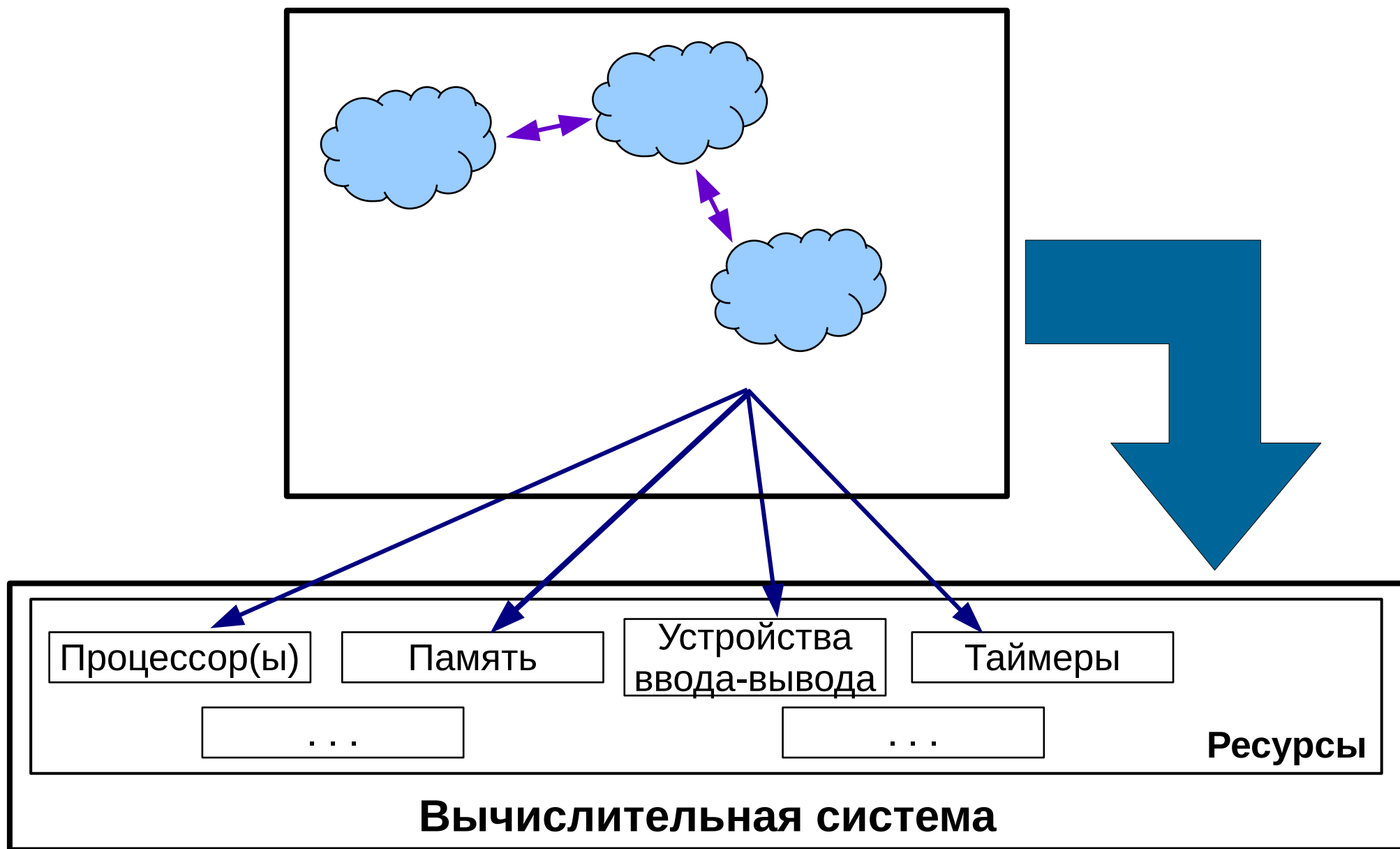






Вычислительная система







✓ Преимущества:

- Близкое к оптимальному использование



✓ Преимущества:

- Близкое к оптимальному использование

✓ Недостатки:

- Одно устройство — свой набор правил



✓ Преимущества:

- Близкое к оптимальному использование

✓ Недостатки:

- Одно устройство — свой набор правил
- Управление на низком уровне

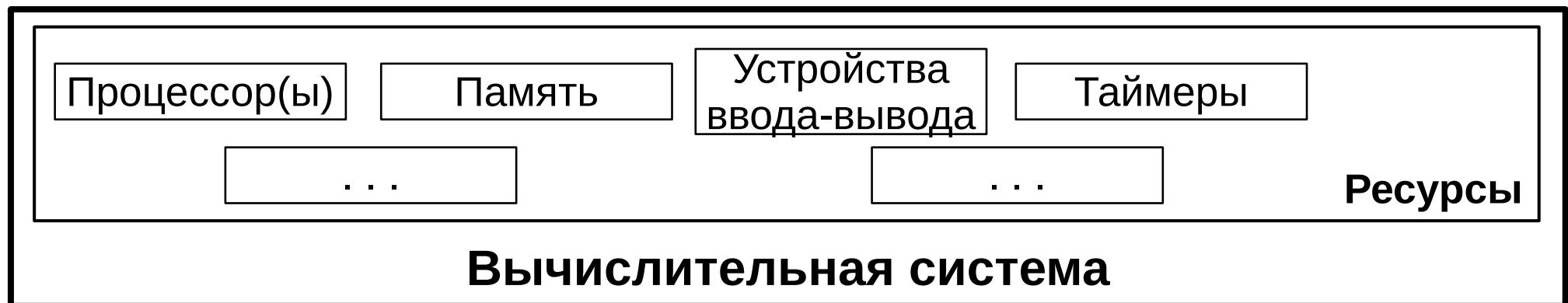


✓ Преимущества:

- Близкое к оптимальному использование

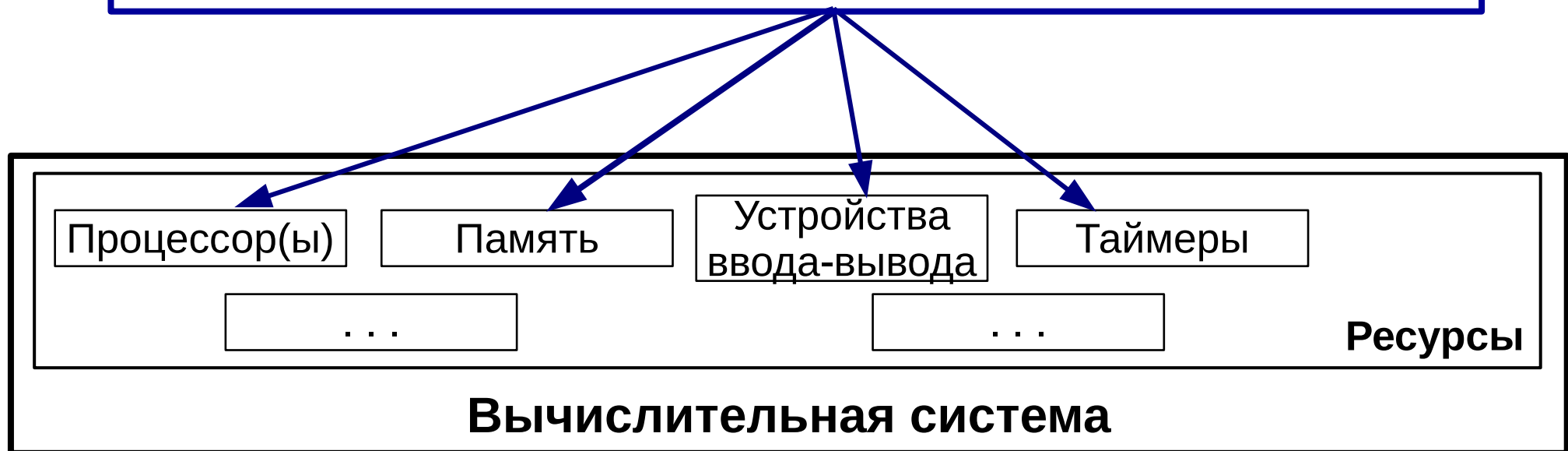
✓ Недостатки:

- Одно устройство — свой набор правил
- Управление на низком уровне
- Жесткая привязка к вычислительной системе



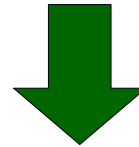


Операционная система

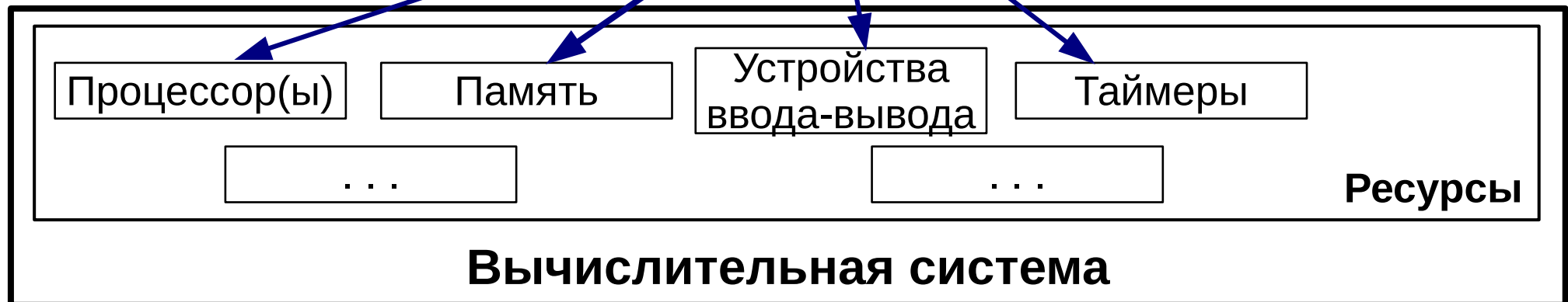




Программная система



Операционная система

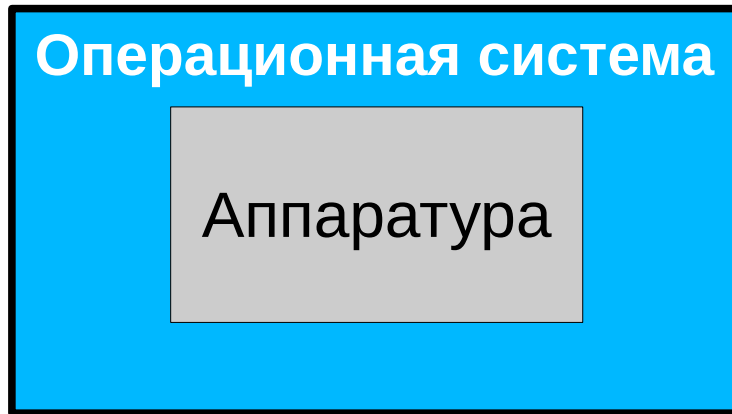




Операционная система – набор программ, которые обеспечивают возможность использования аппаратуры компьютера. При этом аппаратура предоставляет «сырую» вычислительную мощность, а задача ОС заключается в том, чтобы сделать аппаратуру доступной и удобной для пользователей, а также обеспечить рациональное и координированное управление ей для достижения высокой производительности.



Виртуальная машина

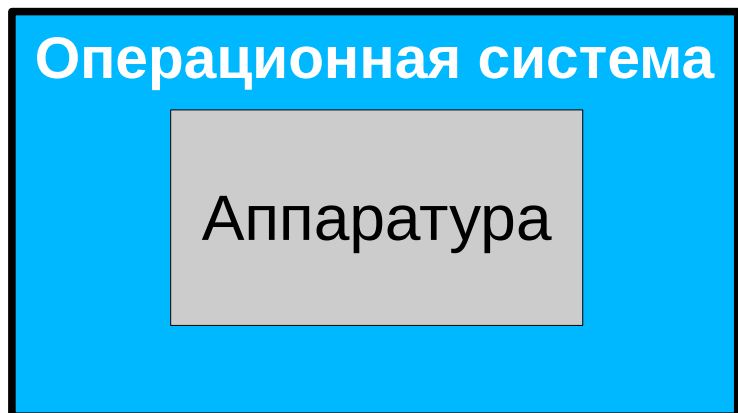




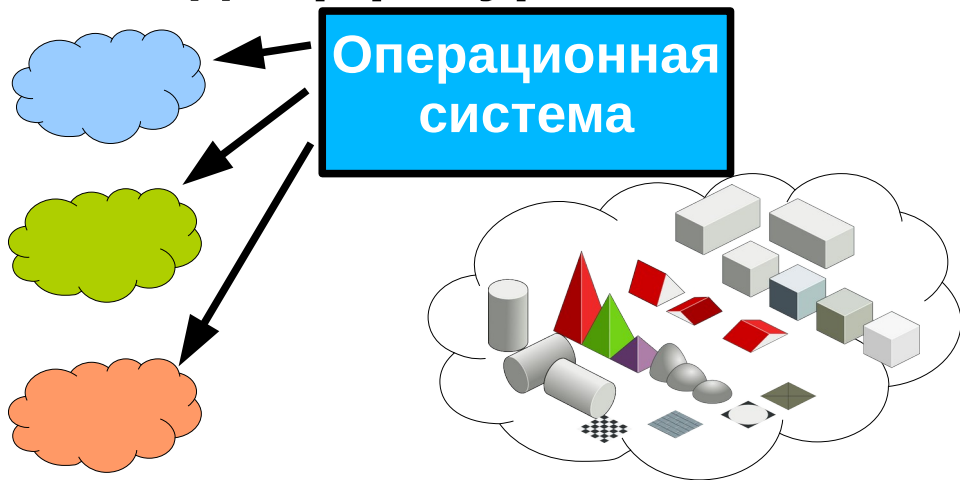
Роли операционной системы



Виртуальная машина



Менеджер ресурсов

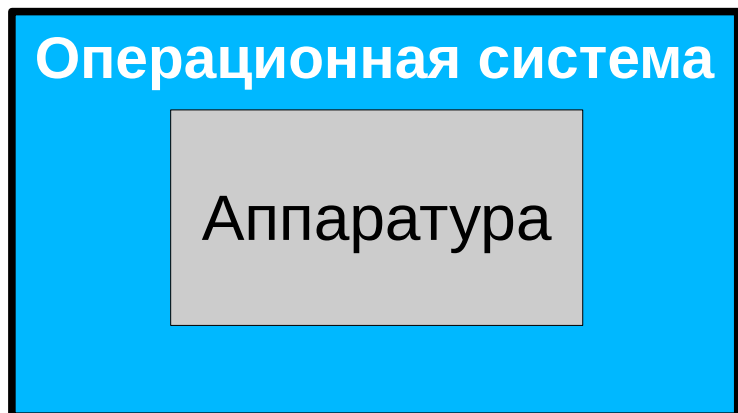




Роли операционной системы



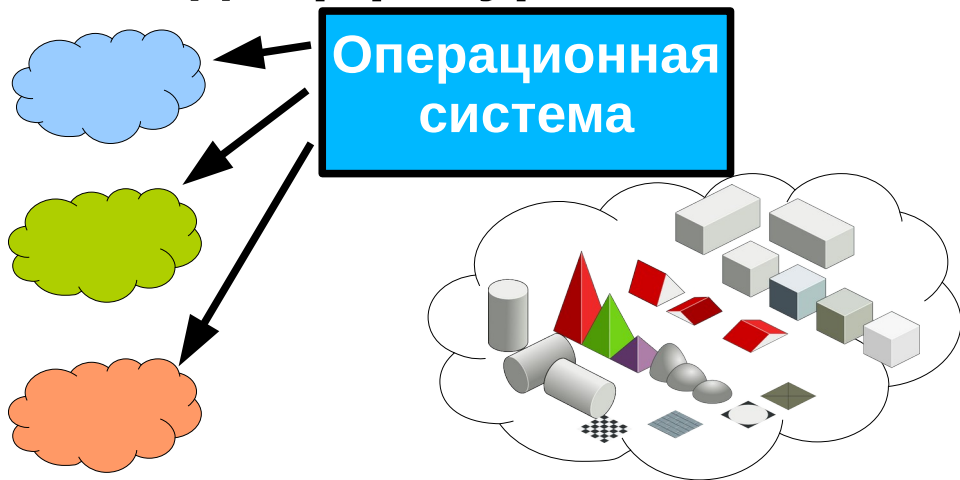
Виртуальная машина



Защитник пользователей и программ



Менеджер ресурсов

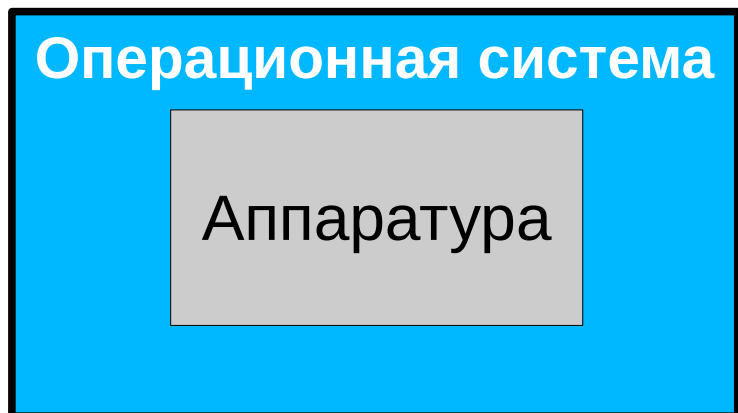




Роли операционной системы



Виртуальная машина



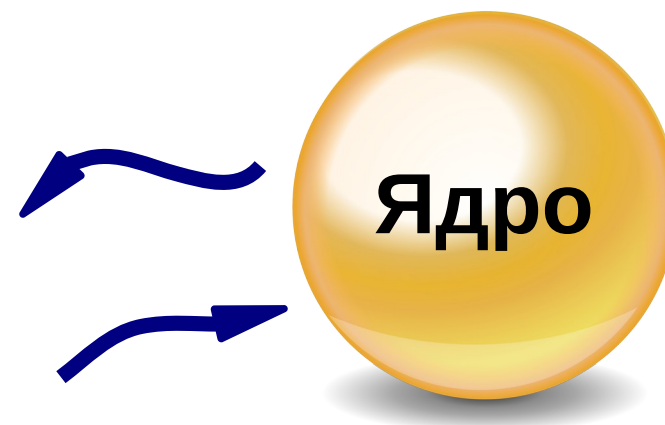
Защитник пользователей и программ



Менеджер ресурсов



Постоянно функционирующее ядро





✓ Преимущества:

- Управление на **высоком уровне абстракции**



✓ Преимущества:

- Управление на высоком уровне абстракции
- **Аппаратно-независимая реализация**



✓ Преимущества:

- Управление на высоком уровне абстракции
- Аппаратно-независимая реализация
- **Изменения в hardware → изменения только ОС**



✓ Преимущества:

- Управление на высоком уровне абстракции
- Аппаратно-независимая реализация
- Изменения в hardware → изменения только ОС

✓ Недостатки:

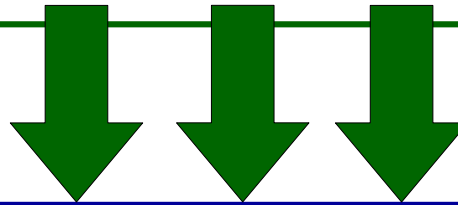
- Накладные расходы на поддержку операционной системы



Программная система



Программная система

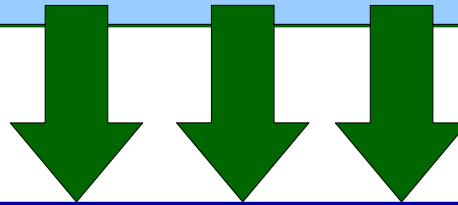


Операционная система



Программная система

Промежуточные компоненты



Операционная система



Промежуточные компоненты — часть программной системы, предназначенная для связи с операционной системой



Задача разработчика программной системы

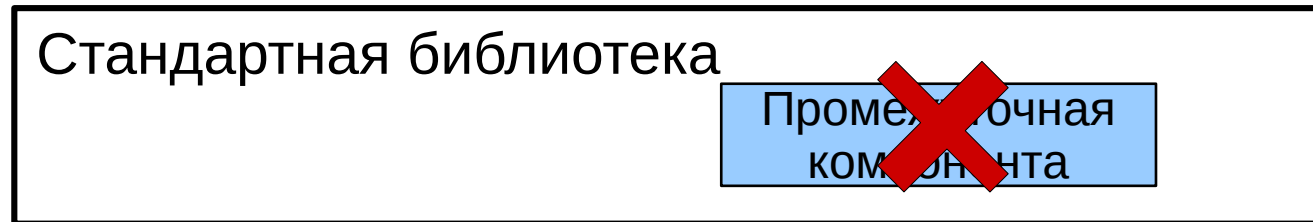
Стандартная библиотека

Промежуточная
компонента





Задача разработчика программной системы

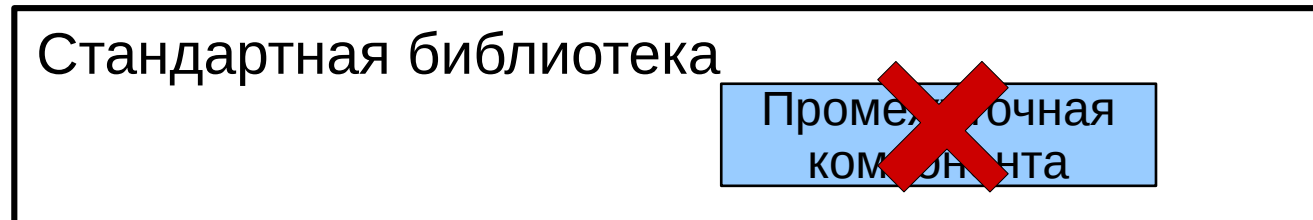


✓ Готовое решение, близкое к требуемому

- Встраивание в программную систему
- Доработка с учетом конкретных требований



Задача разработчика программной системы



✓ Готовое решение, близкое к требуемому

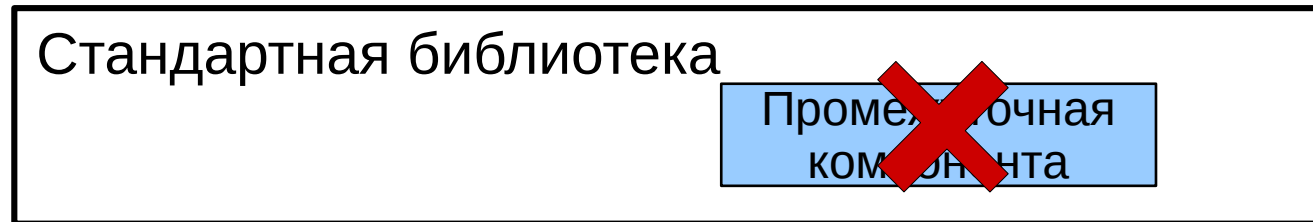
- Встраивание в программную систему
- Доработка с учетом конкретных требований

✓ Конструирование из базовых компонент

- Выбор необходимых компонент
- Объединение в виде композитного элемента



Задача разработчика программной системы



✓ Готовое решение, близкое к требуемому

- Встраивание в программную систему
- Доработка с учетом конкретных требований

✓ Конструирование из базовых компонент

- Выбор необходимых компонент
- Объединение в виде композитного элемента

✓ Оригинальная разработка нового типа



Системное программирование

Системные
ВЫЗОВЫ

Библиотека
С

Компилятор
С



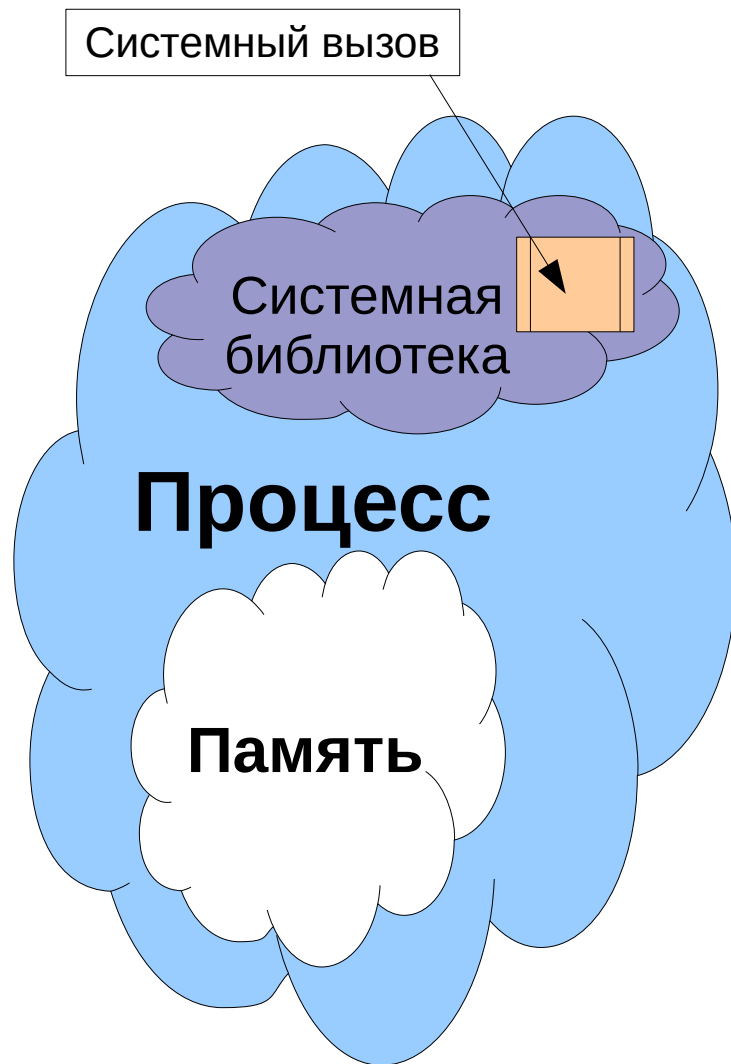
Системное программирование

Системные
ВЫЗОВЫ

Библиотека
С

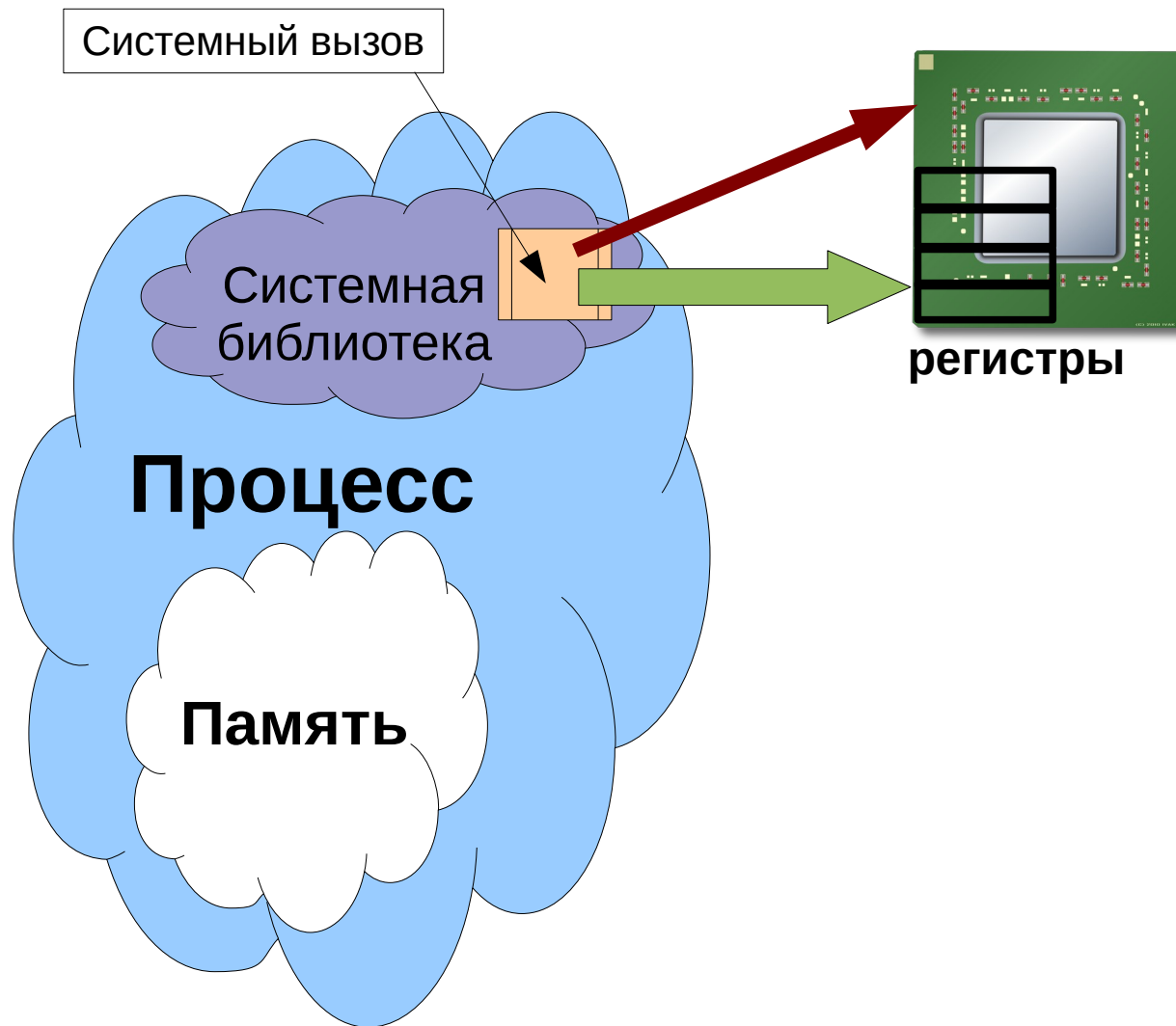
Компилятор
С

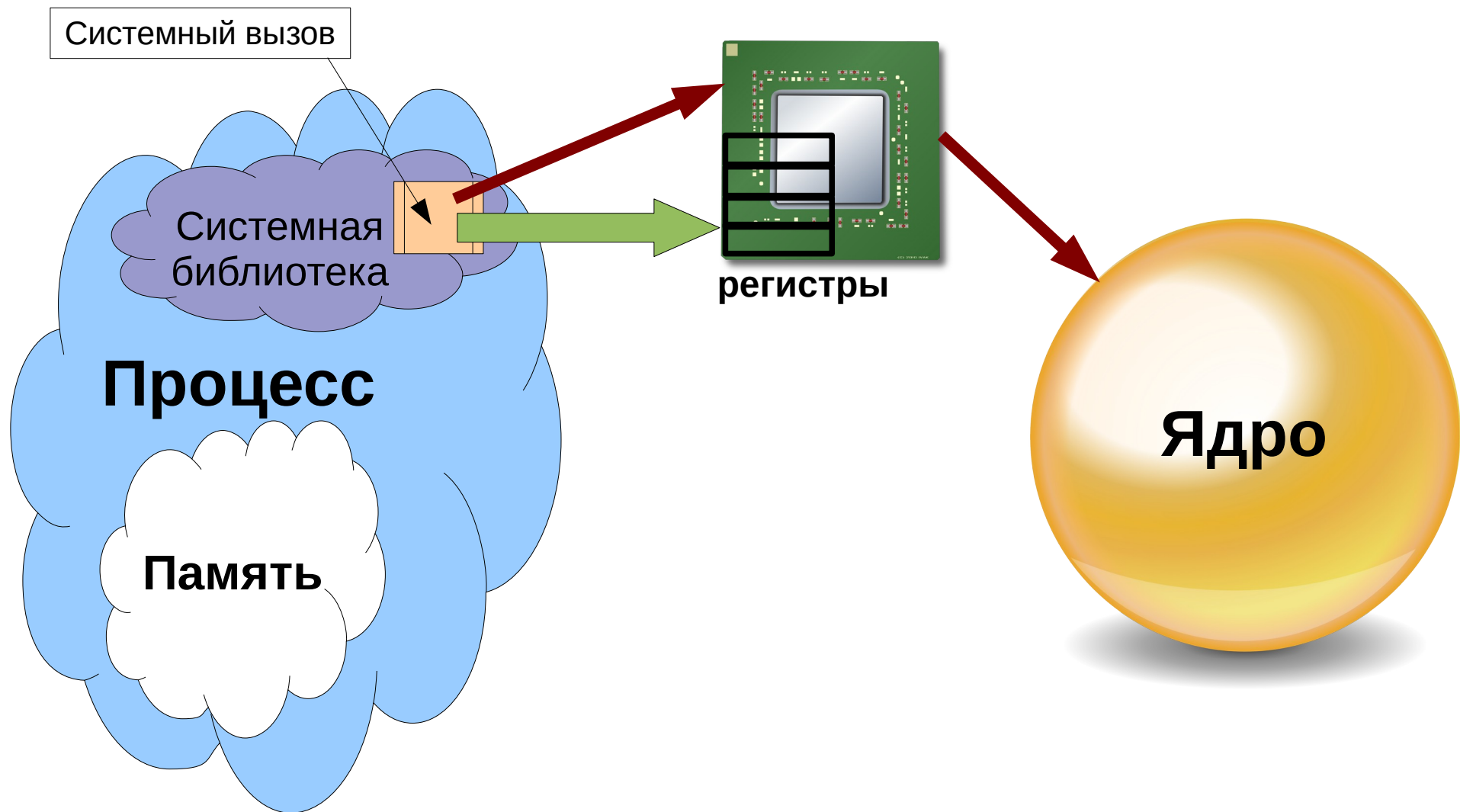
**Системный вызов — интерфейс между ядром
и пользовательской программой**





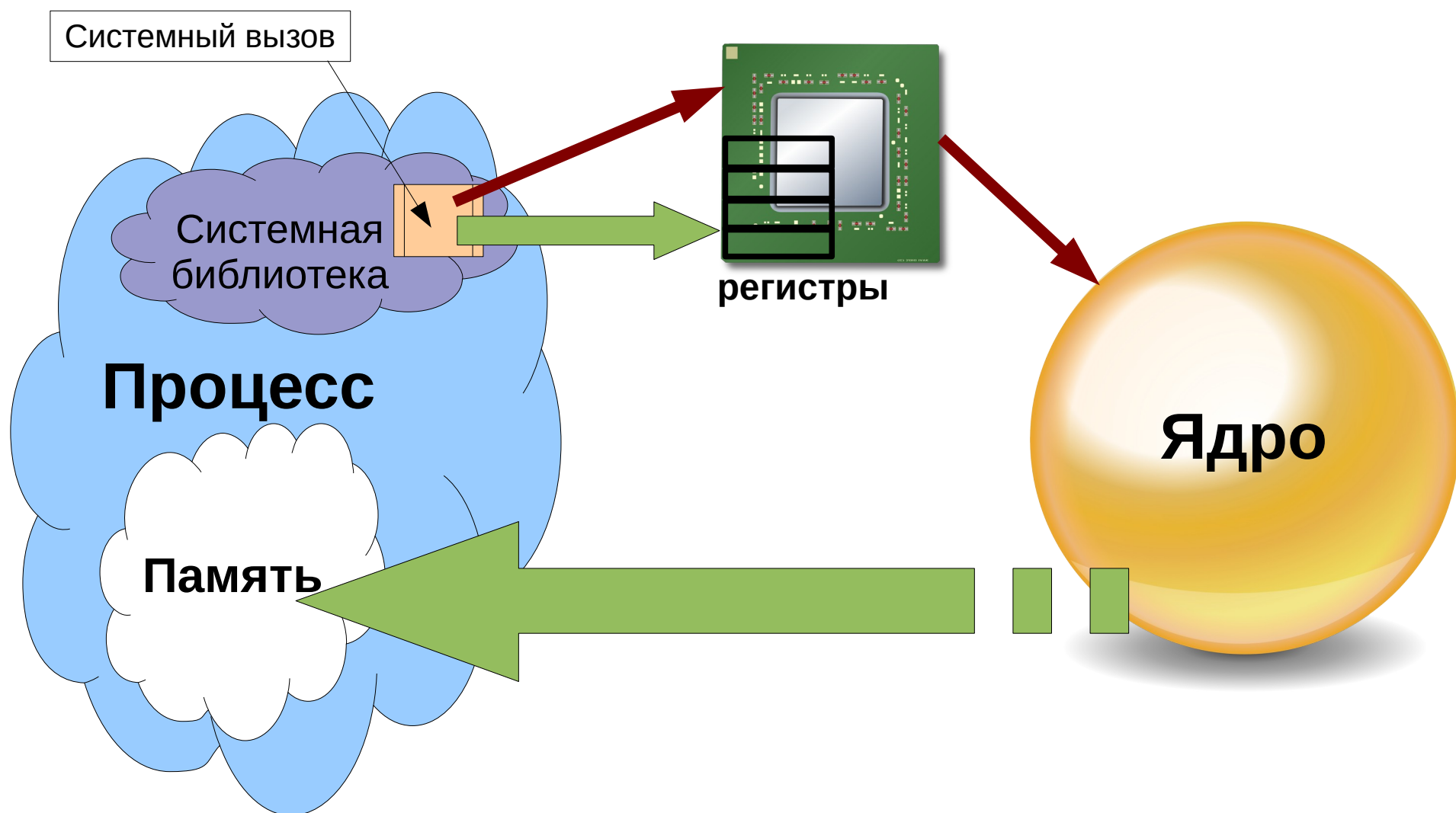
Системные вызовы







Системные вызовы





Пример системного вызова



```
#include <stdlib.h>
```

```
void stupid();
```



Пример системного вызова



```
#include <stdlib.h>
```

```
void stupid();
```

```
int main() {
```

```
    stupid();
```

```
    return 0;
```

```
}
```




Пример системного вызова



```
#include <stdlib.h>
```

```
void stupid();
```

```
int main() {
```

```
    stupid();
```

```
    return 0;
```

```
}
```

```
void stupid() {
```

```
    exit(1);
```

```
    return;
```

```
}
```



Промежуточные компоненты