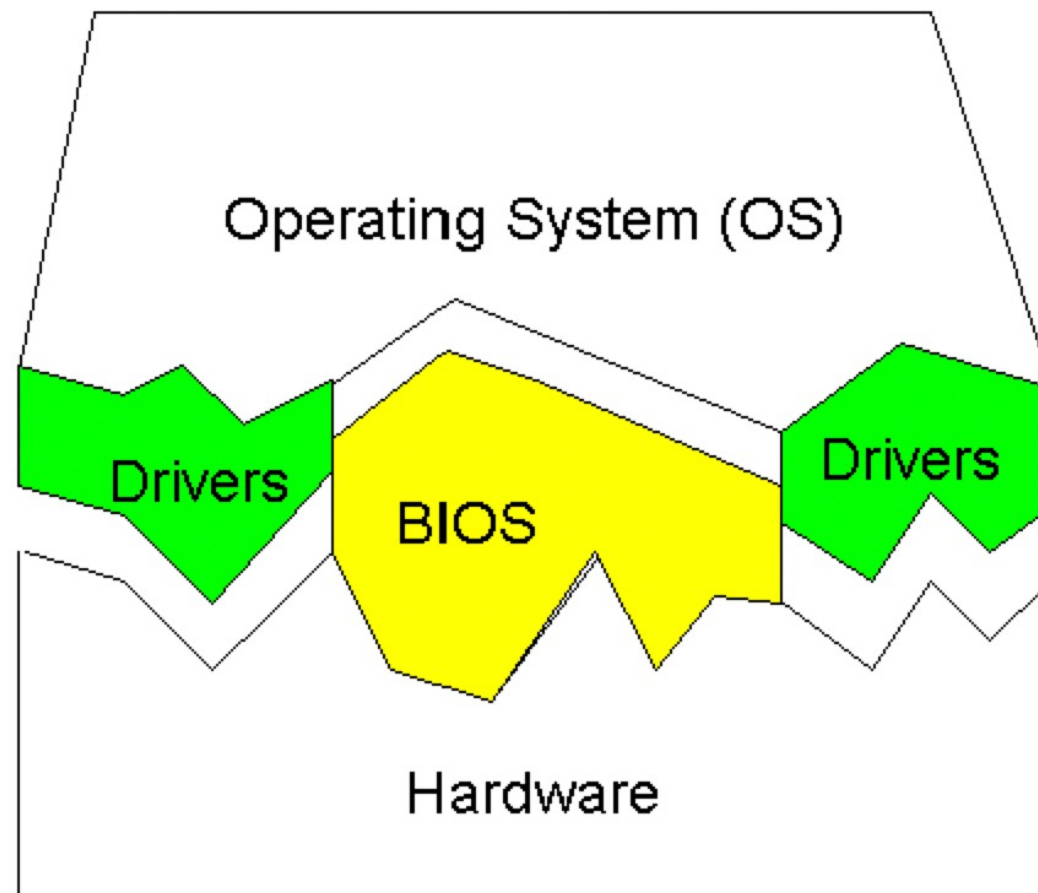


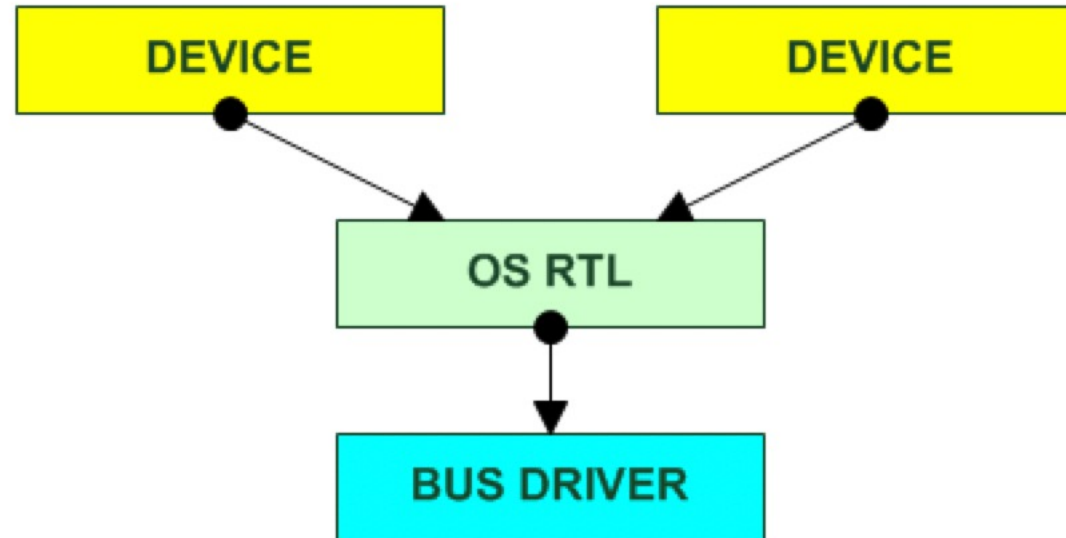
Устройства

Виды устройств и их взаимодействие с ОС и программами пользователя. DMA.

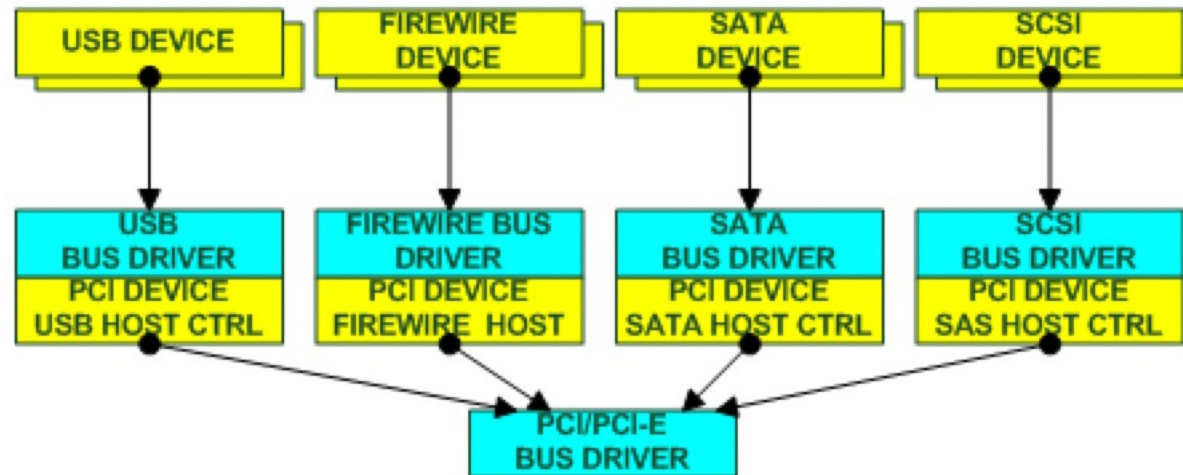
Драйверы



Конфигурирование шин

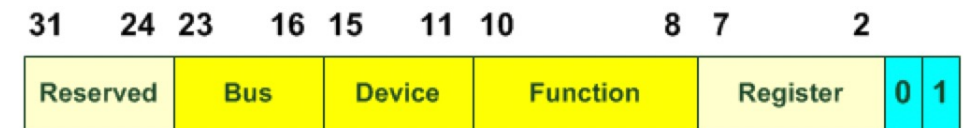


Конфигурирование PCI



Формат адресного пространства PCI

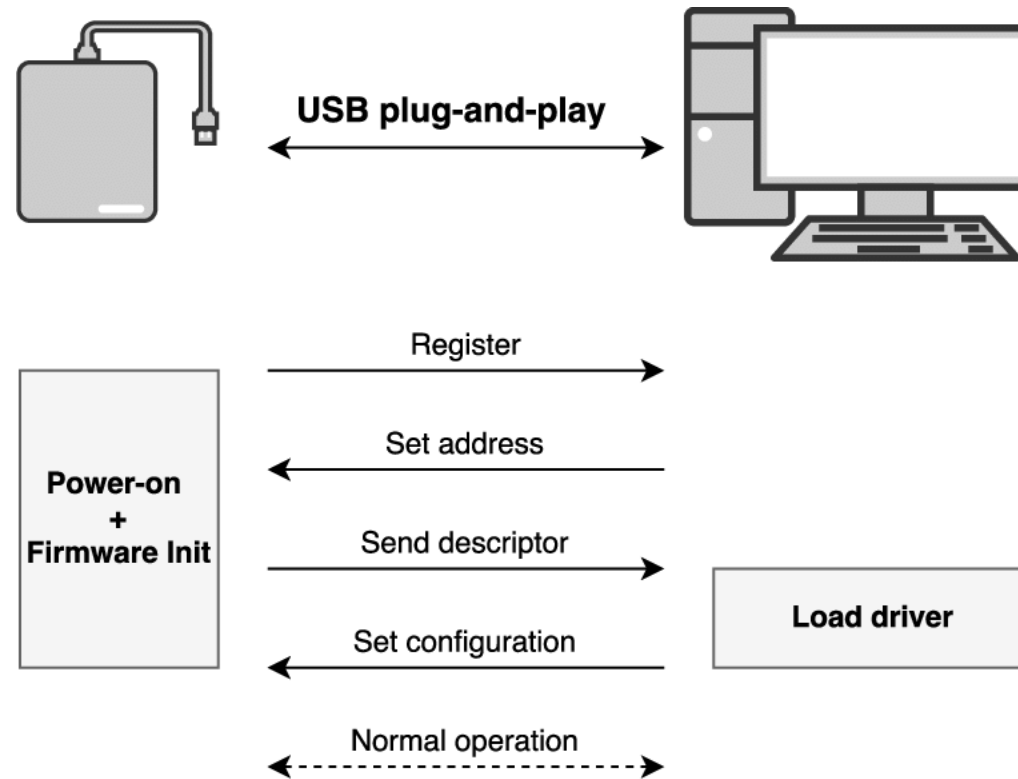
Vendor ID		Device ID	
Command		Status	
Revision ID	Class Code	Subclass code	Interface
Cache Line Size	Latency Time	Header Format	BIST Result
Base Address [0]			
...			
Base Address [5]			
CISPtr			
Subsystem Vendor ID		Subsystem SysID	
ROM Address			
CLOff			
Interrupt Line	Interrupt Pin	Min BusMaster	Max Latency
Vendor specific			



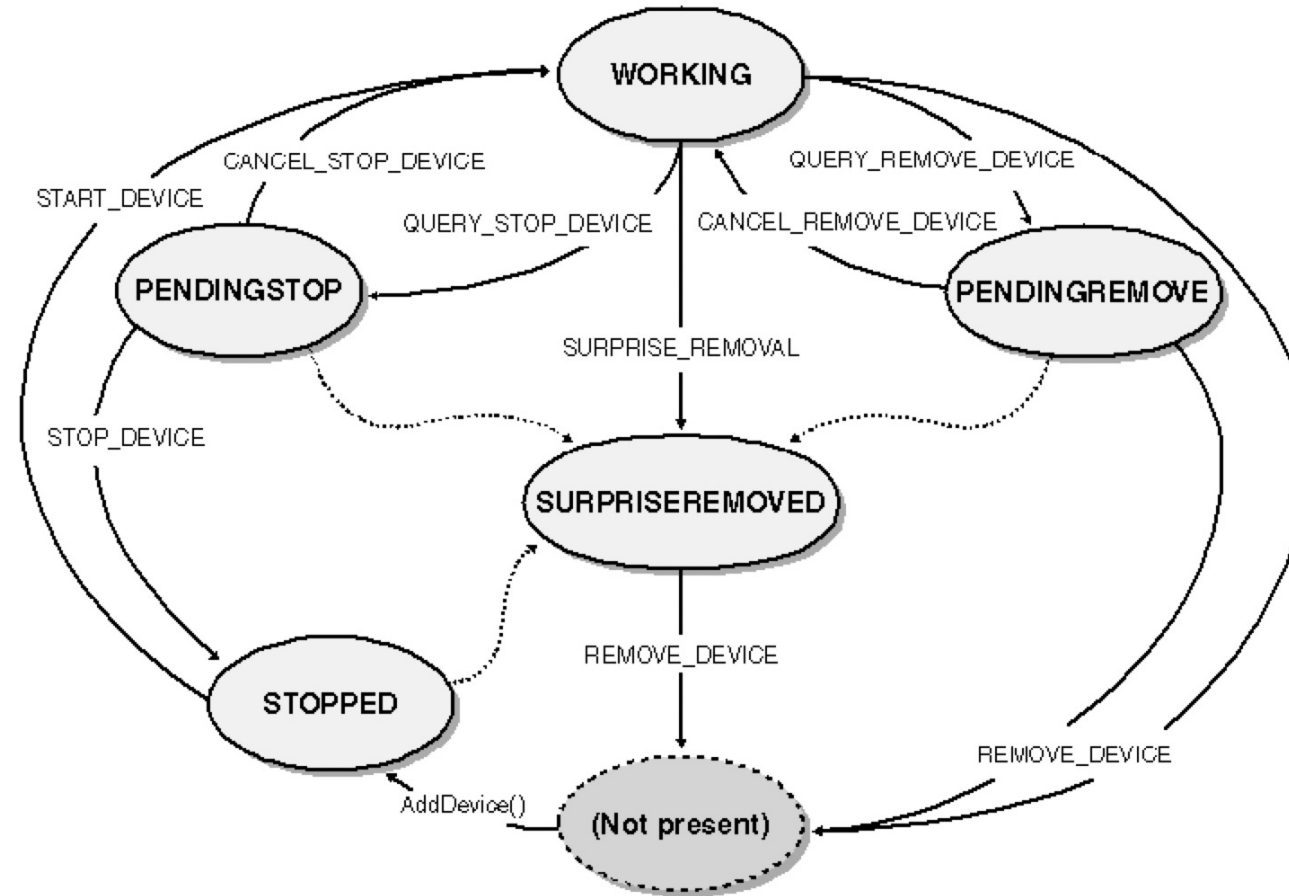
Конфигурирование USB

Descriptor Length = 12h	Descriptor Type = 01h Device
USB Specification, Which comply	
Class	Subclass
Protocol	Max Packet Size
Vendor ID	
Device ID	
Device Release Number	
Manufacturer String Index	Product String Index
Serial Number String Index	Number of configurations

PnP конфигурирование

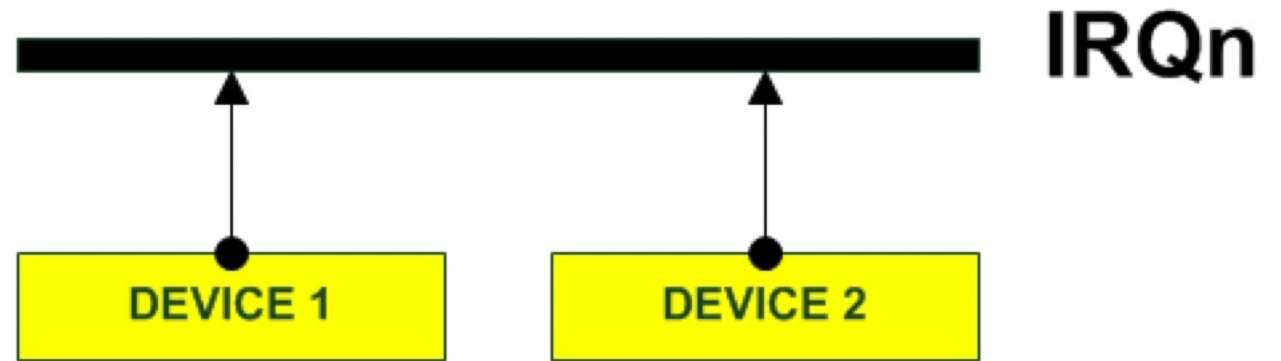


PnP управление устройствами

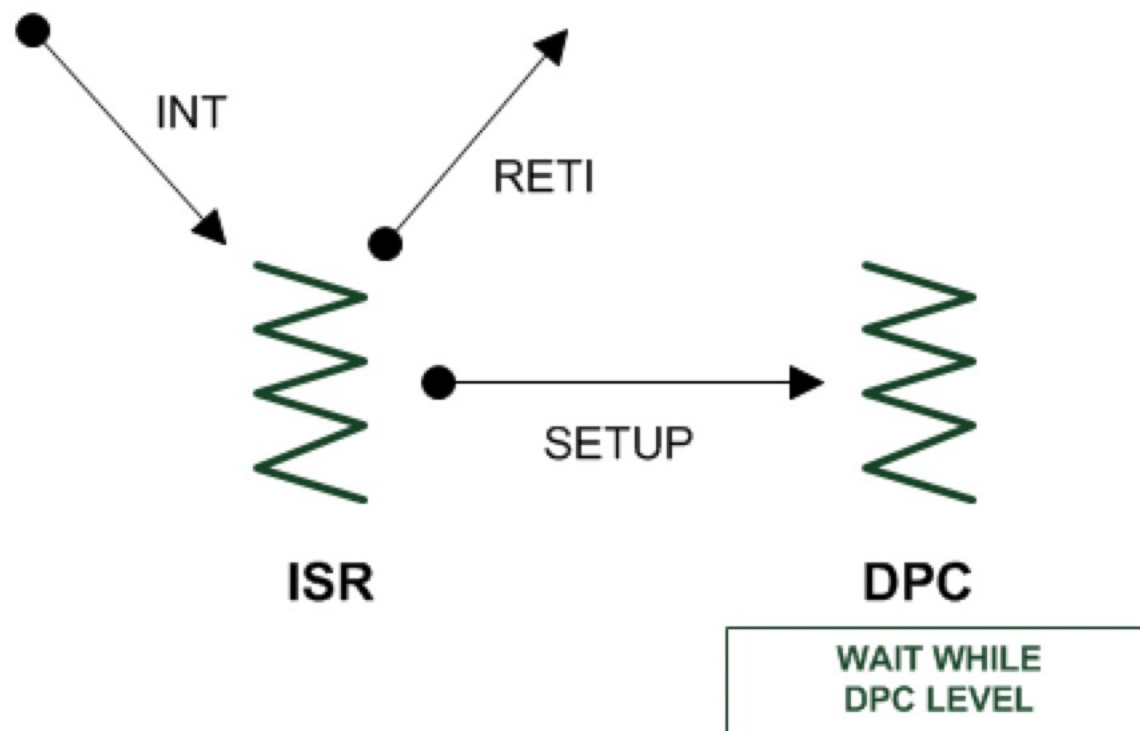


Работа драйверов с прерываниями

- IPC-INTERRUPT
- HW-INTERRUPT
- PAGING
- APC



DPC функции



Синхронизация с обработчиками прерываний

C++

```
KIRQL KeAcquireInterruptSpinLock(  
    _Inout_ PKINTERRUPT Interrupt  
);
```

C++

```
BOOLEAN KeSynchronizeExecution(  
    [in, out]      PKINTERRUPT      Interrupt,  
    [in]           PKSYNCHRONIZE_ROUTINE SynchronizeRoutine,  
    [in, optional] __drv_aliasesMem PVOID SynchronizeContext  
);
```

Unix

- Символьные
- Блочные
- Пространство ядра
- Пространство пользователя

СИМВОЛЬНЫЕ

- `foo_read()`
- `foo_write()`

Блочные

- Буферная память
- strategy routine
- асинхронность

Пространство пользователя

- Один процесс
- Приватное виртуальное пространство
- Полноценное управление исполнением
- Закрытый исходный код

Пространство пользователя vgalib

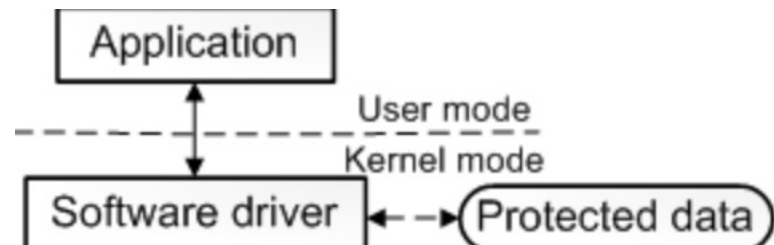
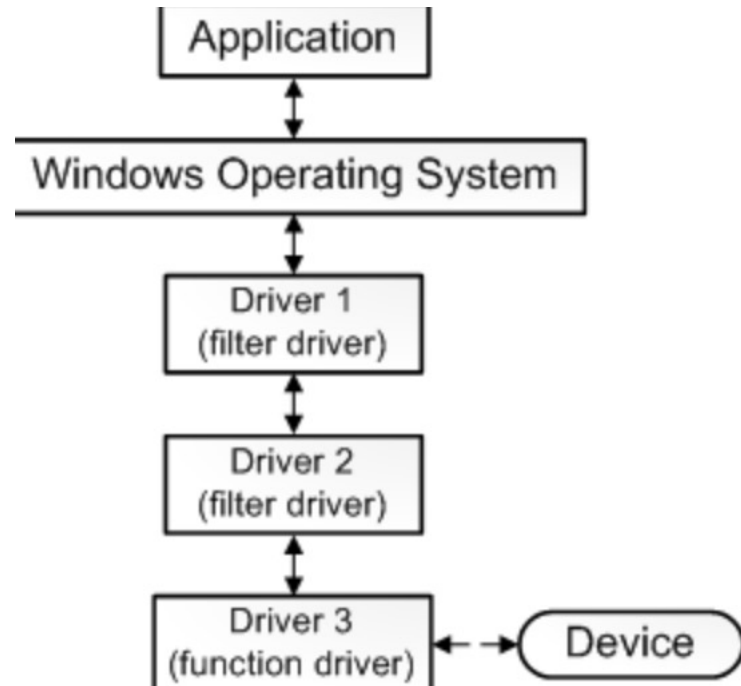


```
1  /* open /dev/mem */
2  if ((mem_fd = open("/dev/mem", O_RDWR) ) < 0) {
3      printf( "VGALib: can' t open /dev/mem \n");
4      exit (-1);
5  }
6
7  /* mmap graphics memory */
8  if ((graph_mem = malloc(GRAPH*SIZE + (PAGE_SIZE-1))) == NULL) {
9      printf( " VGALib: allocation error \n ");
10     exit (-1);
11 }
12 if ((unsigned long)graph_mem % PAGE_SIZE)
13     graph_mem += PAGE_SIZE - ((unsigned long)graph_mem % PAGE_SIZE);
14 graph_mem = (unsigned char *)mmap(
15     (caddr_t)graph_mem,
16     GRAPH_SIZE,
17     PROT_READ|PROT_WRITE,
18     MAP_SHARED|MAP_FIXED,
19     mem_fd,
20     GRAPH_BASE
21 );
22 if ((long)graph_mem < 0) {
23     printf(" VGALib: mmap error \n");
24     exit (-1);
25 }
```

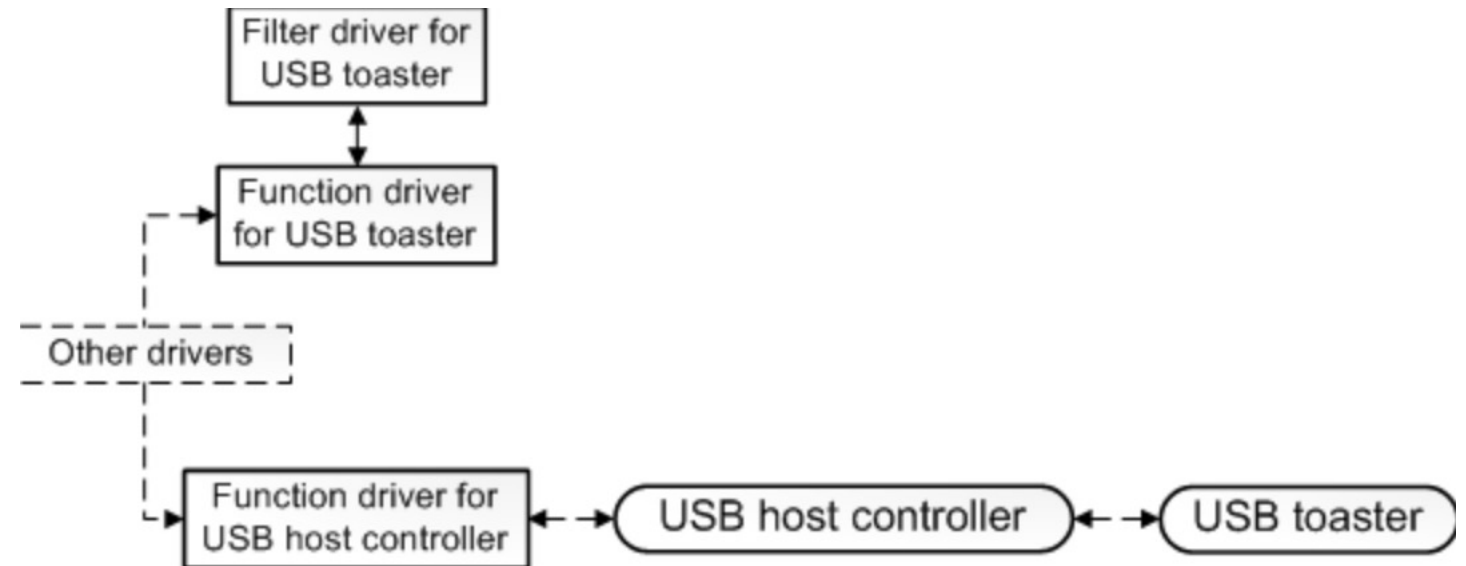

Пространство ядра

- Несколько процессов
- Общие ресурсы
- Прерывания и поочередное опрашивание
- `kmalloc()`

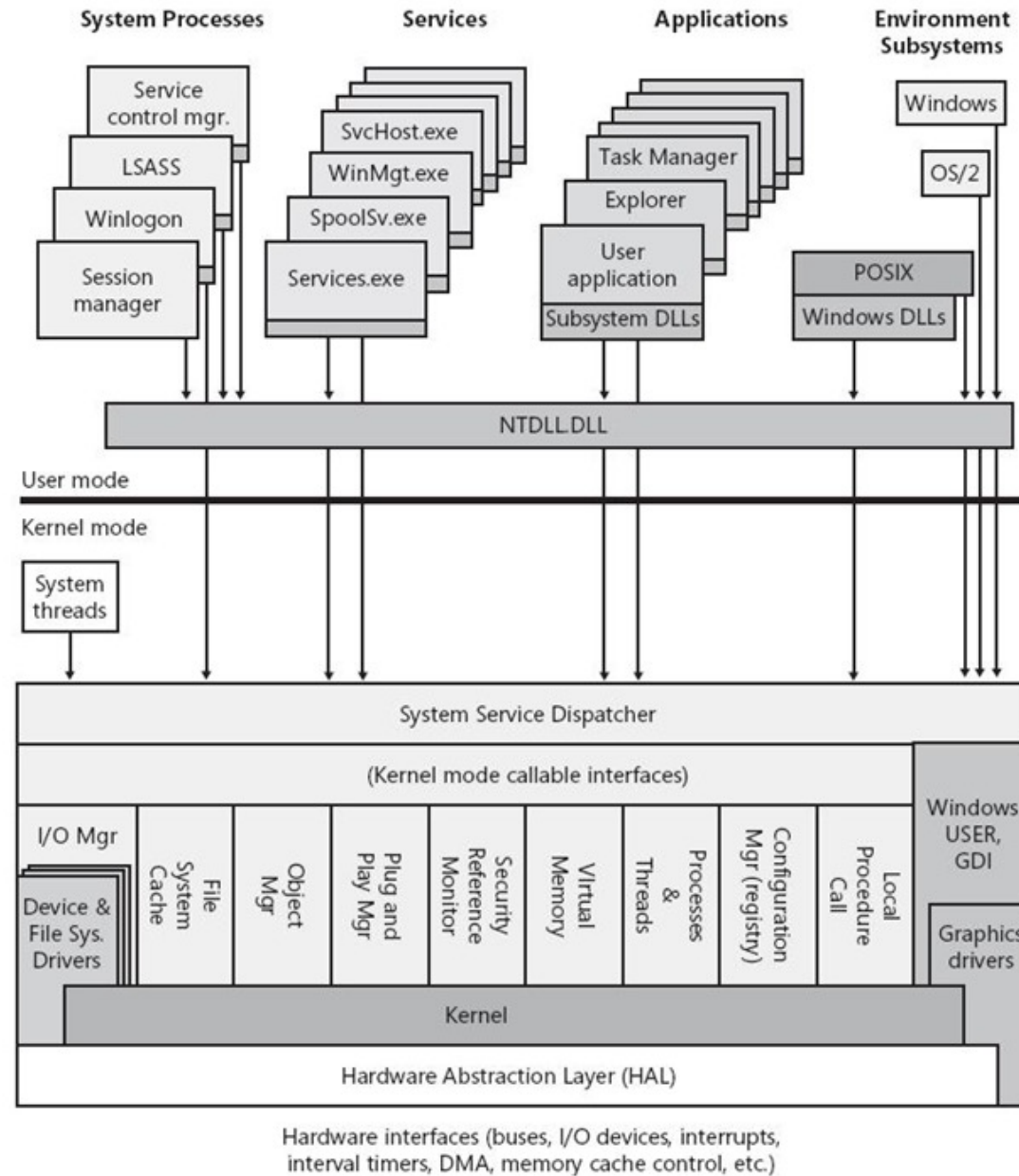
Windows



- Драйвер функции устройства
- Драйвер фильтра устройств
- Драйвер программного обеспечения
- Драйвер фильтра файловой системы
- Драйвер файловой системы

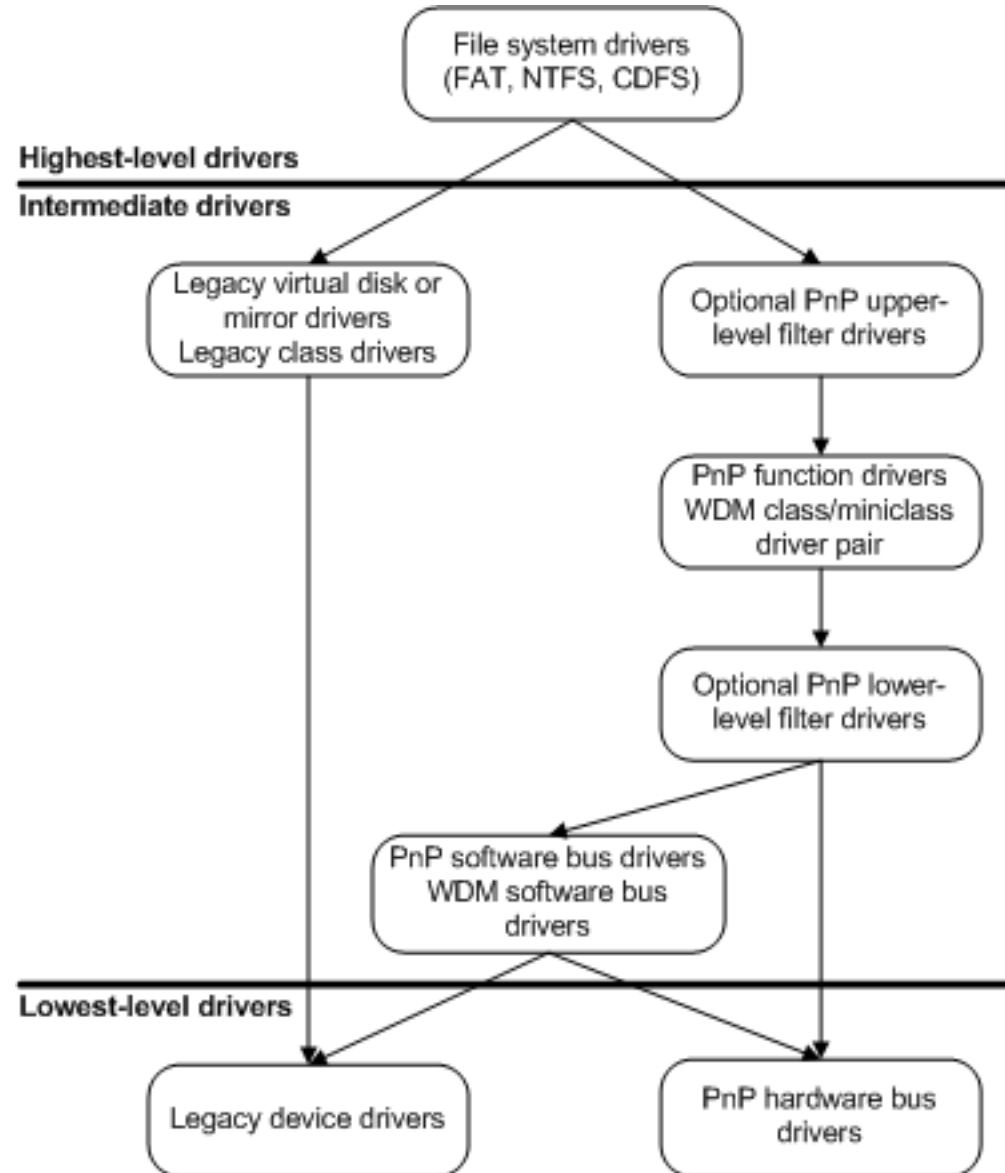


Windows



Windows

- User-mode drivers
- Kernel-mode drivers



Design Goals for Kernel-Mode Drivers

- Портруемость
- Конфигурируется для различных аппаратных и программных платформ.
- Вытеснения и прерывания
- Multiprocessor-safe
- Object-based
- Packet-driven I/O
- asynchronous I/O

WHQL



Storage drivers

блочные устройства разных уровней

- низкого уровня покрывает физическое устройство (ssd,scsi cd)
- высокого уровня покрывает том(несколько дисков)

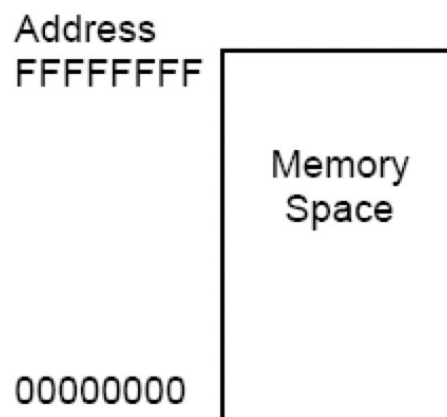
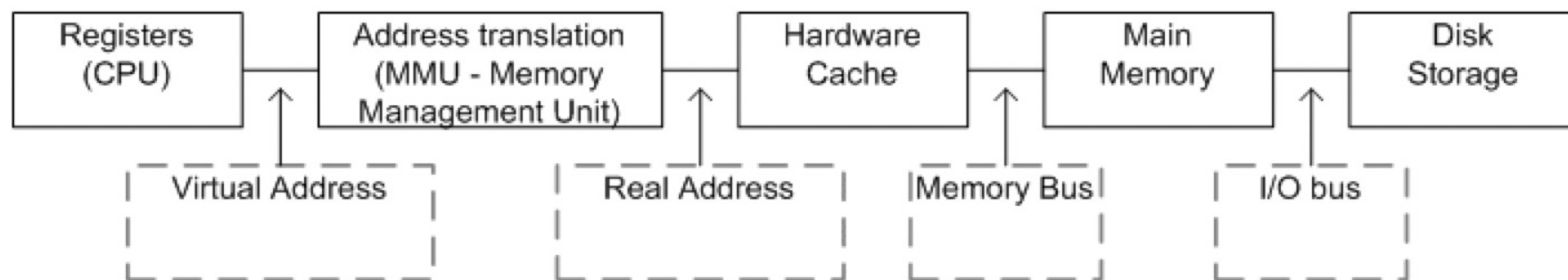
Filesystem drivers

- FSR (File System Recognizer)
- FSD (File System Driver)

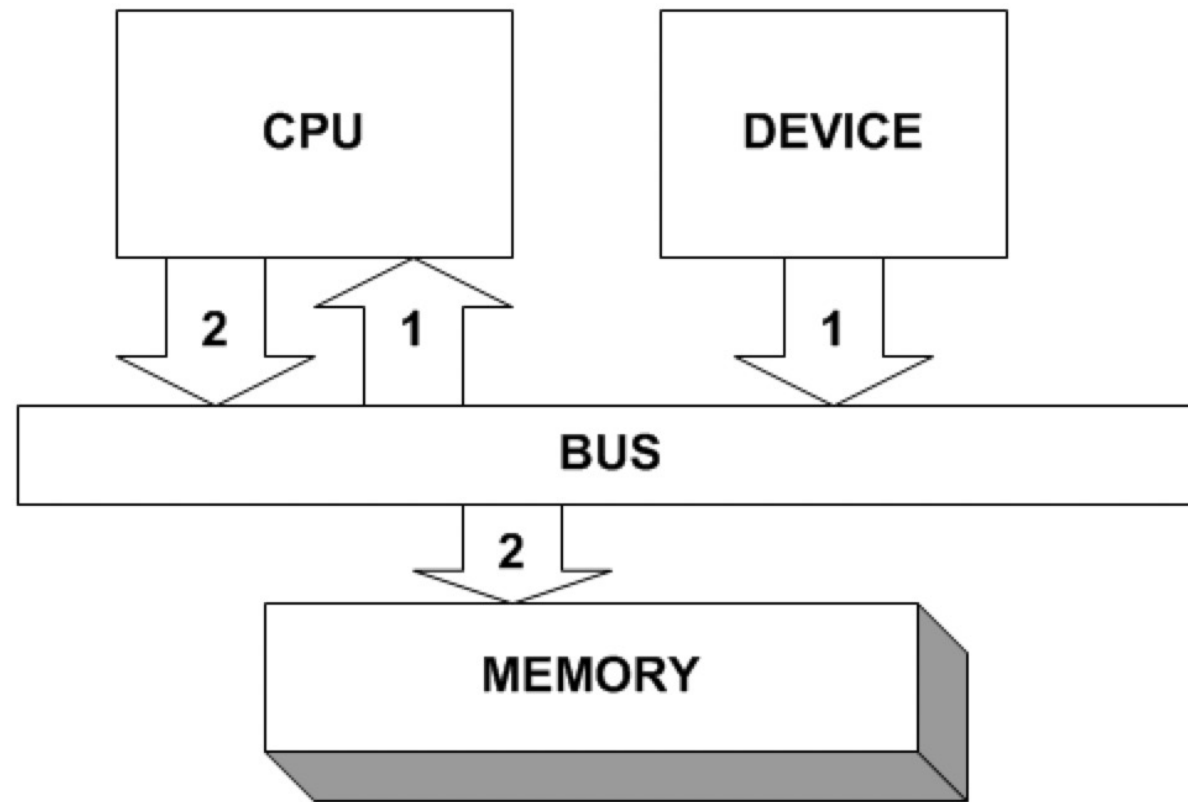
Networks drivers

- Socket Driver
- Network Protocol
- Network Miniport
- Network Intermediate
- Network Filtering Drivers

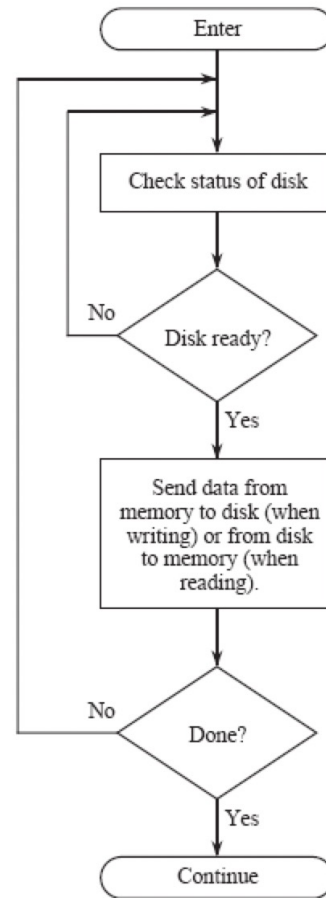
Адресное пространство ввода/вывода



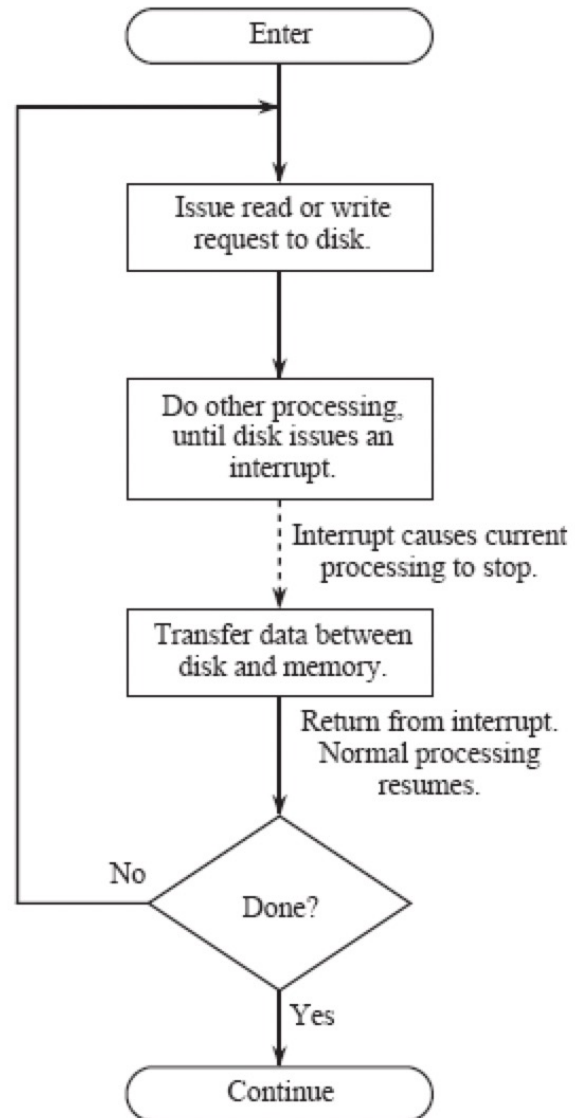
PIO



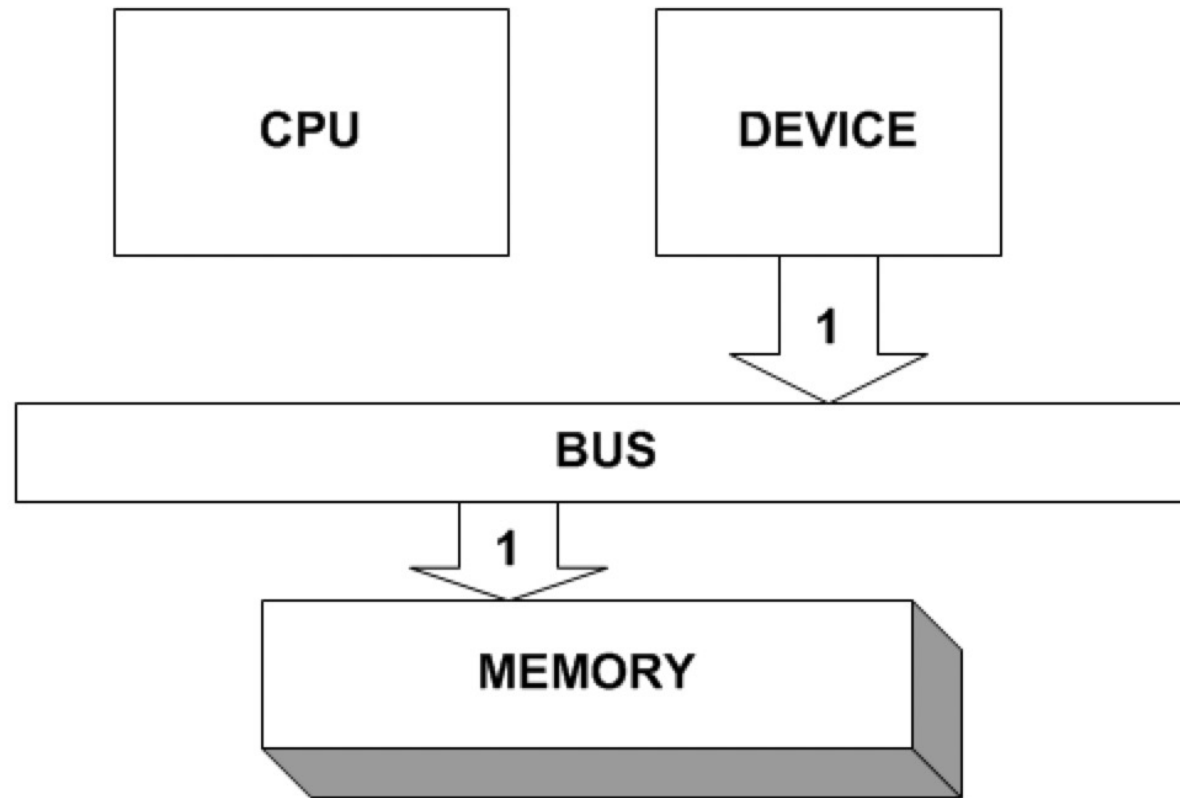
PIO с блокированием



PIO с ожиданием прерывания

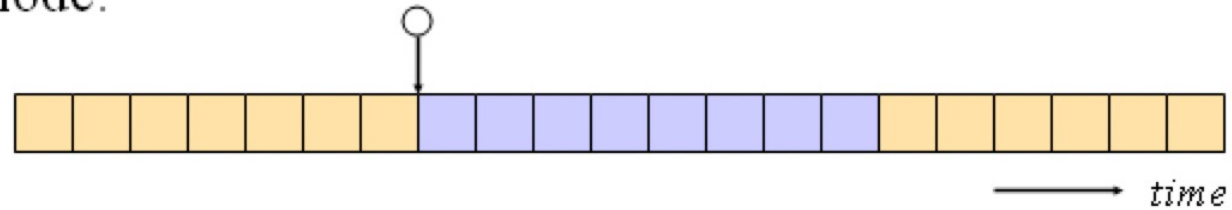


DMA

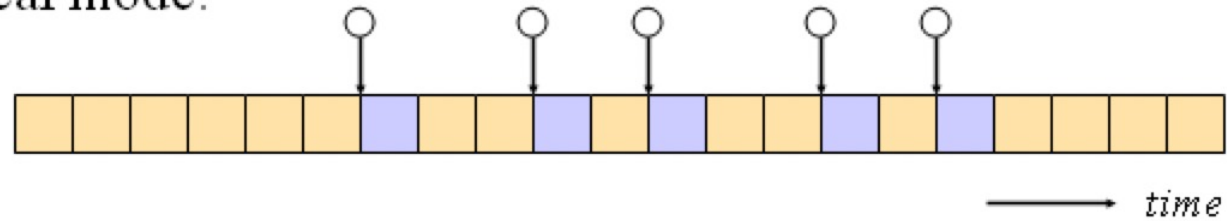


DMA burst

Burst mode:



Cycle steal mode:



Legend:

■ CPU cycle

■ DMA cycle

○ $\overline{BR}/\overline{BG}/\overline{BGACK}$ sequence

Bus mastering DMA

- Любое устройству может захватить шины
- Самостоятельно выставляет на шину сигналы адреса и управления
- Доступ ЦП к шине при этом кратковременно блокируется
- Нет контроллера DMA

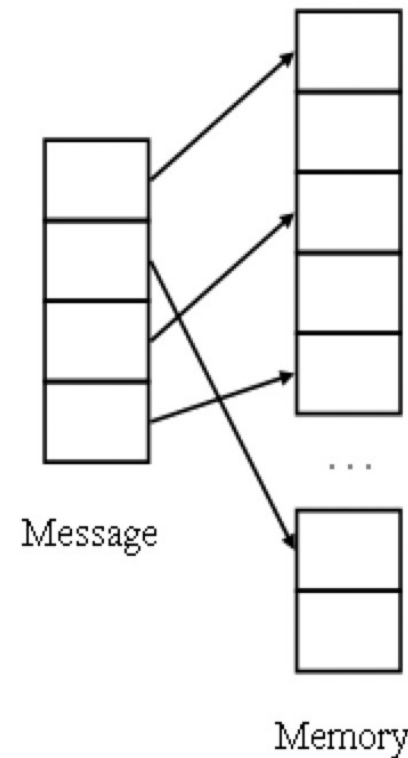
Contiguous DMA

- Выделяется один буфер в оперативной памяти.
- Физический адрес этого буфера записывается в регистр устройства.
- Контроллер устройства инициирует DMA трансфер.
- После того, как буфер полностью заполнен, контроллер устройства инициирует прерывание
- Драйвер операционной системы обрабатывает прерывание, и передает полученные данные из буфера, далее по стеку устройств операционной системы.

Scatter/Gather DMA

1. Выделение подряд идущей физической памяти в ядре операционной системы и промежуточное копирование всех данных туда/оттуда *bounce buffer*.
2. Разбиение операции на подоперации по границам элементов *SGL*, с прерыванием в конце каждой операции..
3. Поддержка *SGL* самим устройством

Scatter (Receive)



Gather (Send)

