

Учебный курс

Основы операционных систем

Лекция 12

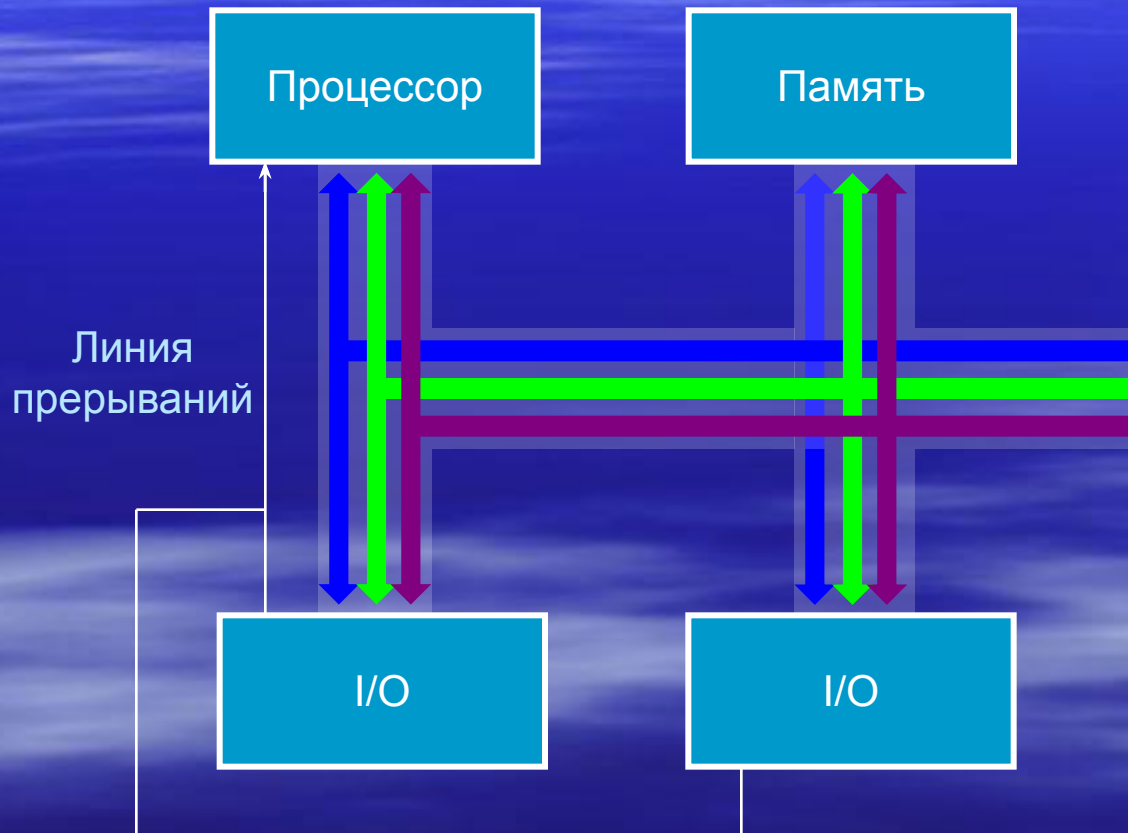
кандидат физико-математических наук, доцент

Карпов Владимир Ефимович

Вывод данных на внешнее устройство

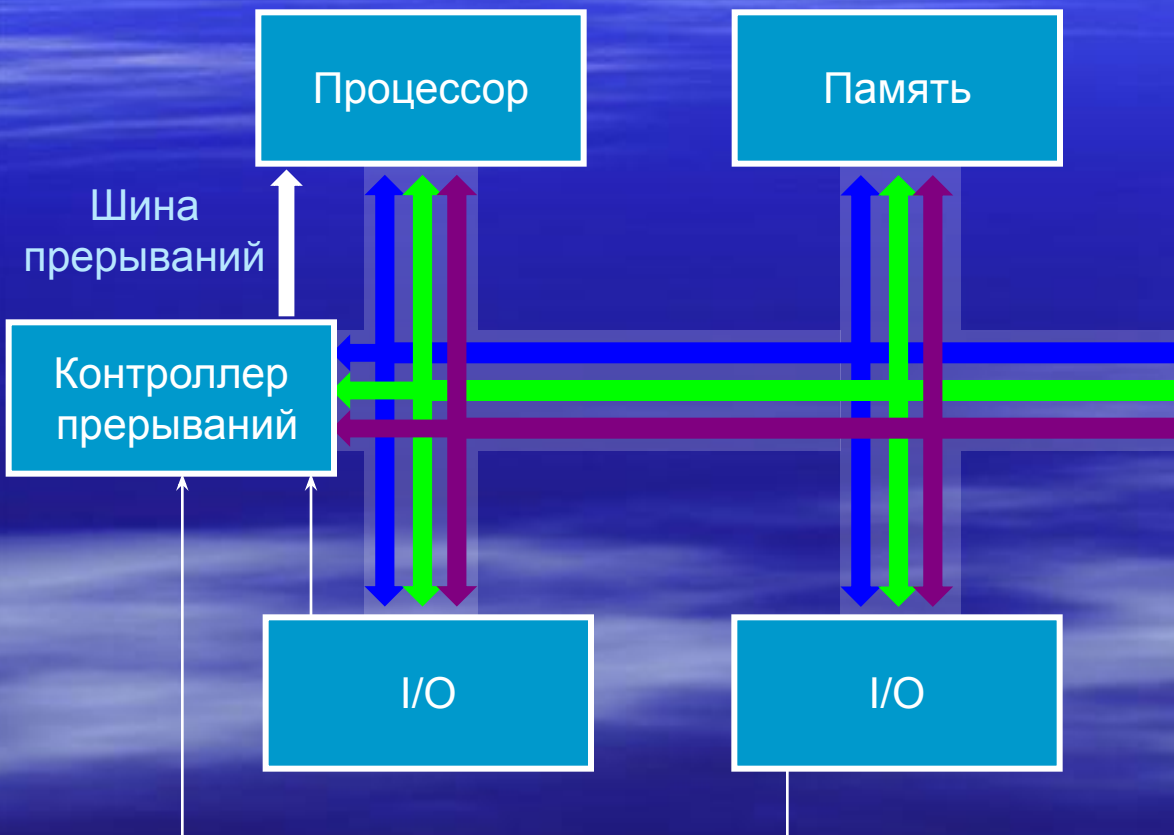


Вывод данных на внешнее устройство



1. После выполнения команды процессор обнаруживает сигнал на линии прерываний
2. Сохраняет часть регистров
3. Передает управление по заранее определенному адресу
4. Обрабатывает прерывание
5. Восстанавливает контекст

Вывод данных на внешнее устройство



1. После выполнения команды процессор обнаруживает сигнал на линии прерываний
2. Сохраняет часть регистров
3. Передает управление по заранее определенному адресу
4. Обрабатывает прерывание
5. Восстанавливает контекст

Внешние прерывания, исключительные ситуации и программные прерывания

Внешние прерывания

- Обнаруживаются процессором *между* выполнением команд
- Сохраняется часть контекста перед выполнением *следующей* команды
- *Не связаны* с работой процессора и *непредсказуемы*

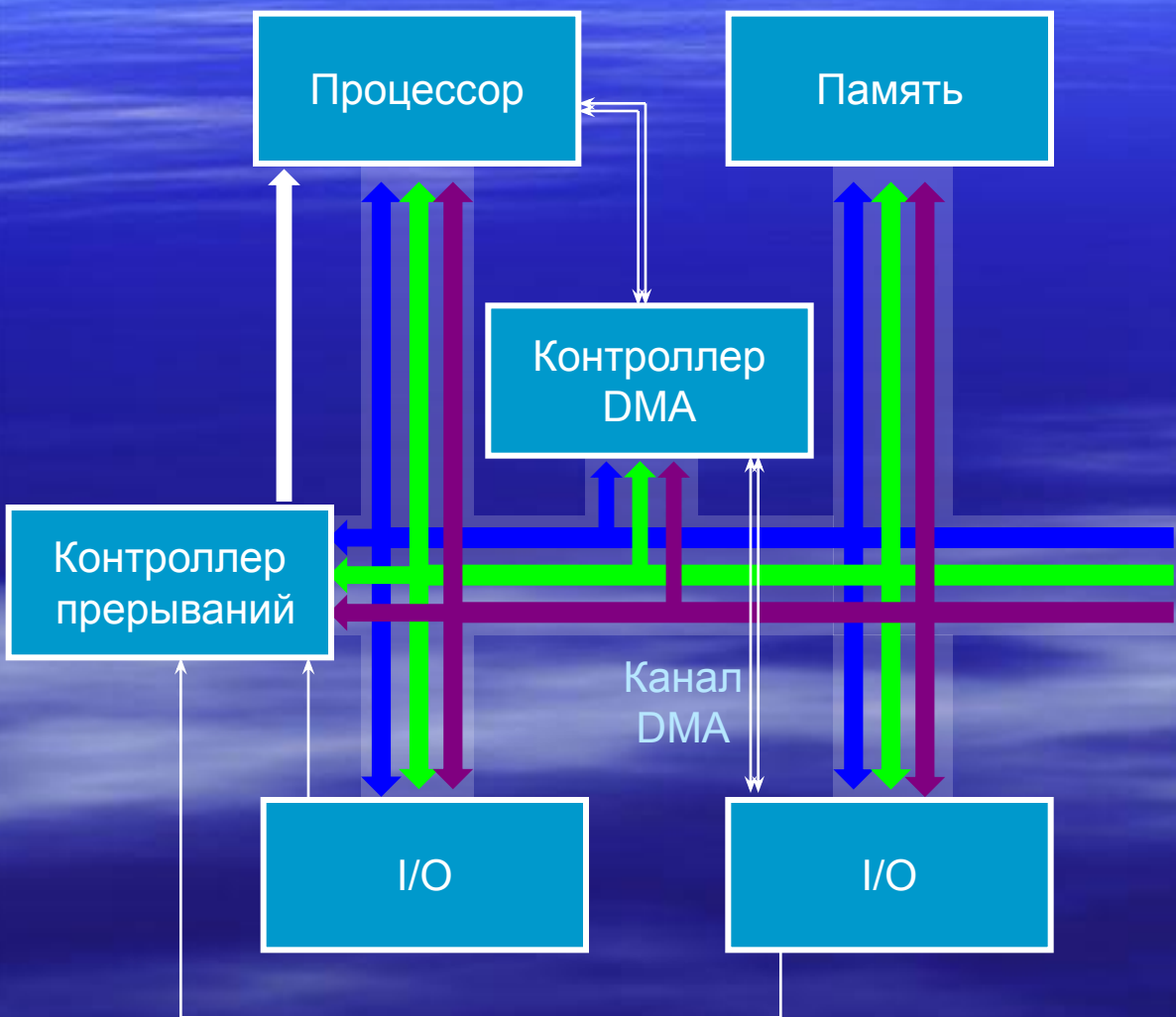
Исключительные ситуации

- Обнаруживаются *во время* выполнения команды
- Сохраняется часть контекста перед выполнением *текущей* команды
- *Связаны* с работой процессора, но *непредсказуемы*

Программные прерывания

- Происходят *в результате* выполнения команды
- Сохраняется часть контекста перед выполнением *следующей* команды
- *Связаны* с работой процессора и *предсказуемы*

Прямой доступ к памяти (Direct memory access – DMA)



1. Контроллер DMA программируется
2. После получения сигнала от устройства I/O запрашивает у процессора управление магистралью
3. Получив управление, выставляет адрес и извещает устройство I/O
4. Используя шины данных и управления совместно с устройством I/O передает информацию
5. Возвращает управление магистралью

Основные направления различия устройств ввода-вывода

- Скорость обмена информацией (от нескольких байтов до нескольких Гигабайтов в секунду)
- Возможность использования несколькими процессами параллельно
- Запоминание выведенной информации для последующего ввода
- Символьные и блочные
- Только для ввода информации, только для вывода информации и read-write устройства

Структура системы ввода-вывода



Систематизация внешних устройств

- Символьные устройства (клавиатура, модем, терминал и т.д.)
- Блочные устройства (магнитные и оптические диски и ленты и т.д.)
- Сетевые устройства (сетевые карты)
- Все остальные (таймеры, графические дисплеи, видеокамеры и т.д.)

Интерфейс между базовой подсистемой ввода-вывода и драйверами

Символьные устройства

Ввести символ – get

Вывести символ – put

Выполнить произвольную команду – ioctl

(Re)инициализировать драйвер и устройство – open

Временно завершить работу с устройством – close

Остановить работу драйвера – stop

Опросить состояние устройства – poll

Блочные устройства

Прочитать блок – read

Записать блок – write

Найти блок – seek

Функции

базовой подсистемы ввода-вывода

- Поддержка блокирующихся, неблокирующихся и асинхронных вызовов
- Буферизация и кэширование входных и выходных данных
- Осуществление spooling'а и монопольного захвата внешних устройств
- Обработка ошибок и прерываний
- Планирование последовательности запросов на выполнение операций ввода-вывода

Блокирующиеся, неблокирующиеся и асинхронные вызовы

- При блокирующемся системном вызове процесс переходит из состояния *исполнение* в состояние *ожидание*. После выполнения операций ввода-вывода в полном объеме он разблокируется.
- При неблокирующемся системном вызове операции ввода-вывода могут быть выполнены неполностью. Процесс либо неблокируется совсем, либо блокируется не более чем на определенное время.
- При асинхронном системном вызове процесс никогда не блокируется. Операции ввода-вывода выполняются в полном объеме.

Буферизация и кэширование

Причины буферизации в базовой подсистеме ввода-вывода

- Разные скорости приема и передачи информации участников обмена
- Разные объемы данных, которые могут быть приняты или переданы участниками обмена одновременно
- Необходимость копирования данных из приложения в ядро ОС и обратно

Буфер – область памяти для запоминания информации при обмене данными между устройствами, процессами или между устройством и процессом

Буферизация и кэширование

Разница между кэшем и буфером

- Буфер служит для согласования параметров участников обмена информацией и для ее промежуточного хранения. Кэш применяется для ускорения доступа к данным.
- Кэш всегда содержит копию данных, существующих где-либо еще. Буфер часто содержит единственный экземпляр данных в системе.

Кэш (cache) – область быстрой памяти, содержащая копию данных, расположенных где-либо в более медленной памяти, предназначенная для ускорения работы вычислительной системы

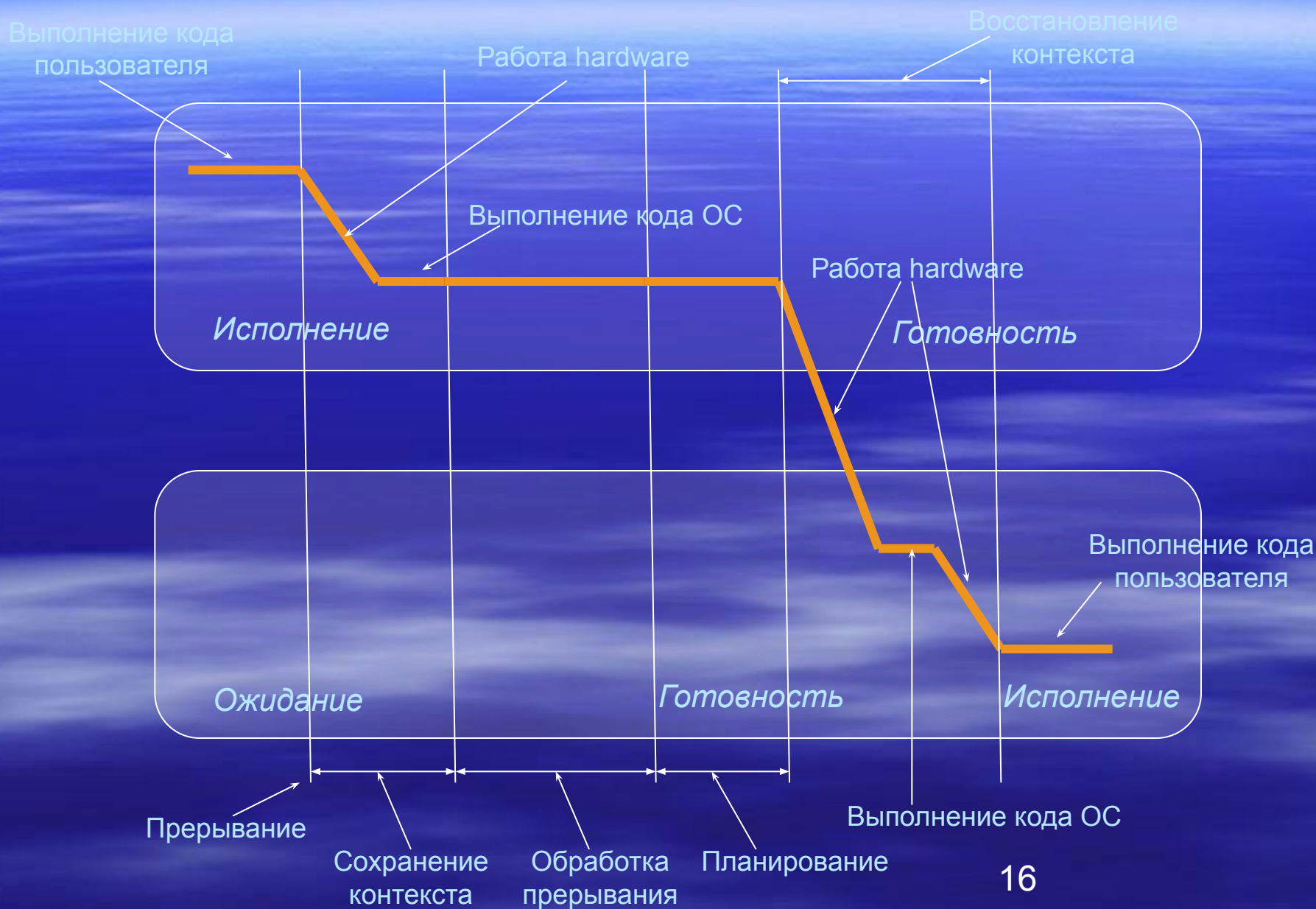
Spooling и захват устройств

Способы использования неразделяемых устройств

- Монопольный захват устройства.
- Spooling.

Spool – буфер, содержащий входные или выходные данные для устройства, на котором следует избегать чередования его использования различными процессами

Обработка прерываний и ошибок



Обработка прерываний и ошибок

Действия операционной системы

- Определение устройства, выдавшего прерывание.
- Взаимодействие с устройством.
- Проверка успешности выполнения операции.
- Попытка устранения возможных ошибок.
- Определение процесса, ожидающего этого прерывания. Перевод его из состояния *ожидание* в состояние *готовность*.
- Если есть еще процессы с неудовлетворенными запросами к этому устройству – инициализация нового запроса.

Действия по обработке прерывания и компенсации ошибок могут быть частично делегированы драйверу устройства – функция `intr` в интерфейсе драйвера