

ФИО \_\_\_\_\_ Группа \_\_\_\_\_ Коллоквиум (заполняется экзаменатором при проверке)

**В ответах указывается № вопроса и дается информация, относящаяся исключительно к заданному вопросу. Писать аккуратно, разборчиво.**

| №  | ВОПРОС   | ОТВЕТ  |
|----|--|--|
| 1. | В модельной схеме обработки прерываний обсуждалось включение режима блокировки прерывания. На какой стадии обработки прерывания включается данный режим? Почему необходимо включение данного режима (обосновать)?  | Включается на аппаратной стадии. Информация о точке текущего прерывания (адрес прерывания, часть регистров и.т.п.) автоматически размещается в специальной регистровой памяти процессора. Если в это время придет другое прерывание, то информация о точке текущего прерывания будет потеряна, что означает, что мы не сможем продолжить выполнение процесса с прерванного места. Блокировка прерывания необходима для того, чтобы избежать подобных потерь. |
| 2. | Предположим в ОС UNIX System V заблокировано кэширование обменов с блок-ориентированными устройствами. Какое предельное количество непроизводительных обменов (дополнительных обменов) может произойти при чтении блока N файла (считаем, что файл уже открыт). Кратко обосновать. | Три. Система адресации блоков файлов в индексном дескрипторе использует тройную косвенную адресацию  |
| 3. | Описать алгоритм формирования списка занятых блоков файловой системы <b>fs5</b> .  | Запускаем цикл по области индексных дескрипторов. Считываем очередной ID. Проверяем содержимое поля «ссылки на данный ID каталогов файловой системы». Если это поле равно нулю (это означает, что ID свободен) переходим к следующему ID. В противном  |

|    |   |  |
|----|---|--|
|    |   | случае. Выбираем и добавляем в список <b>занятых</b> блоков файловой системы номера блоков, занятых данным файлом. Последовательно просматриваем 13 элементов, описывающих адресацию блоков файла (до завершения): номера блоков с прямой адресацией (10 шт.), номера блоков организованных с косвенной адресацией 1,2 и 3-х уровневой . |
| 4. | В системе используется трехуровневая таблица страниц процесса. Перечислить все возможные «непроизводительные» действия при обращении к данной таблице во время выполнения процесса. | Загрузка таблицы второго уровня, загрузка таблицы третьего уровня.   |
| 5. | Прокомментировать, что означает требование атомарности операций семафора <b>Дейкстры</b> ?  | Это означает, что если операция начала выполняться, то ее выполнение не может быть прервано до момента завершения.   |
| 6. | Что определяет использование конкретного коммуникационного домена в сокетах (привести примеры)?   | Использование того или иного коммуникационного домена определяет область видимости (возможности использования) сокета (например, AF_Unix или AF_INET) и правила именования сокетов (например, полное имя файла в случае AF_Unix или IP адрес + номер порта для AF_INET).   |
| 7. | Написать программу на Си, обеспечивающую сортировку по возрастанию значений файла (имя передается через командную строку), содержащего значения типа <b>int</b> .                   | ФАЙЛ не может быть полностью загружен в оперативную память.  |
| 8. | Написать программу на Си, которая обеспечит реализацию: <b>cmd1   cmd2   cmd3</b> , где <b>cmdi</b> – имена доступных исполняемых файлов, передающихся                              | Достаточно написания правильной программы без контроля верности исполнения системных вызовов.  |

|    |   |  |
|----|---|--|
|    | в программу через командную строку  |  |
| 9. | Описать модельную схему установки контрольной точки по адресу <code>Adr</code> в программе средствами <b>ptrace</b> (от чьего имени выполняются действия, в каких состояниях находятся участвующие процессы, какая последовательность действий, сопутствующие данные) | <p>Родительски процесс устанавливает контрольную точку в сыновнем процессе.</p> <ol style="list-style-type: none"> <li>1. Сыновний процесс приостановлен.</li> <li>2. Считываем (посредством <code>ptrace</code>) команду по адресу <code>Adr</code> сегмента кода сына. Сохраняем пару: <code>Adr</code> , команда.</li> <li>3. Записываем в сегмент кода (посредством <code>ptrace</code>) сына по адресу <code>Adr</code> команду, выполнение которой вызовет прерывание и возникновение обрабатываемого сигнала. Например, команду, выполняющую деление на ноль.</li> <li>4. Можем запустить сыновний процесс, например, с точки останова</li> </ol> |