

121. Родительский процесс создал неименованный канал, получив дескрипторы `f[0], f[1]`, после чего породил дочерний процесс. Оба процесса используют канал для двухсторонней передачи данных, при этом каждый из процессов использует дескриптор `f[0]` для чтения, а `f[1]` - для записи. Описанная схема
1. обеспечивает гарантированную двухстороннюю передачу данных;
 - v 2. дает непредсказуемый результат, включая возможные дедлоки;
 3. соответствует спецификации системного вызова `pipe()`;
 4. аналогична схеме использования именованного канала;
123. Можно ли открыть именованный канал на чтение и на запись?
1. да;
 - v 2. нет;
 3. в зависимости от ситуации;
125. RPC расшифровывается как
1. удаленный вызов процедур;
 2. механизмы взаимодействия посредством сокетов;
 3. принцип локальности ссылок;
- v 4. механизмы межпроцессного взаимодействия;
127. Фрагмент программы:
- ```
...
if(fork())
printf("text\n");
...
Каким процессом будет напечатан текст?
v 1. родителем;
2. потомком;
3. родителем и потомком;
```
129. Фрагмент программы:
- ```
...  
if(i=fork()==0)  
printf("text\n");  
...  
Каким процессом будет напечатан текст?  
1. родителем;  
v 2. потомком;  
3. родителем и потомком;
```
131. Фрагмент программы:
- ```
...
if(i=fork()!=0)
printf("text\n");
...
Каким процессом будет напечатан текст?
v 1. родителем;
2. потомком;
3. родителем и потомком;
```
133. Клиентский суррогат (client stub)
1. передает параметры реальной функции сервера;
  - v 2. упаковывает данные в сетевое сообщение;
  3. распаковывает полученные от клиентского процесса данные;
  4. подготавливает файл спецификаций RPC;
135. Файл спецификаций RPC
1. подготавливается клиентским суррогатом;
  2. подготавливается серверным суррогатом;
  - v 3. подготавливается пользователем;
  4. генерируется утилитой `rpcgen`;
137. Ethernet-адрес
1. состоит из четырех байт и используется на физическом уровне;
  2. назначается администратором и состоит из нескольких частей, разделенных точками
  3. состоит из четырех байт и используется на сетевом уровне;
  - v 4. назначается производителем оборудования и является уникальным адресом;
139. 112.255.255.112 – IP-адрес
- v 1. класса A;
2. класса B;
  3. класса C;
  4. класса D;

141. 193.155.55.65 - IP-адрес
- 1. класса А;
  - 2. класса В;
  - v 3. класса С;
  - 4. класса D;
143. Максимальное число хостов в классе В
- 1. 16777214;
  - 2. 254;
  - v 3. 65534;
144. Ядро состоит из
- 1. взаимосвязанных системных таблиц;
  - 2. раздела управляющих структур и машинно-независимой частью программной секции;
  - 3. раздела управляющих структур;
  - 4. раздела управляющих структур и взаимосвязанных системных таблиц;
  - v 5. раздела управляющих структур и программной секции;
145. Процессу, выполняющемуся в системной фазе соответствует
- 1. исполняемый код пользовательской программы;
  - v 2. реентеральный код ядра;
  - 3. один или несколько процессов;
  - 4. ничего из перечисленного неверно;
146. Для синхронизации процессов в системной фазе используется
- 1. аппарат прерываний;
  - v 2. аппарат семафоров (аппарат событий);
  - 3. аппарат блокировки;
  - 4. аппарат сохранения/восстановление контекста пользователя;
  - 5. ничего из перечисленного неверно;
147. Какой из процессов диспетчерский процесс, занимающийся свопингом, выгрузит из ОП в первую очередь при нехватке места для загрузки образа процесса, находящегося во внешней памяти
- 1. процесс, имеющий самый меньший приоритет среди всех;
  - 2. процесс, дольше всех находящийся в ОП;
  - 3. процесс, занимающий в ОП наибольшее место;
  - 4. процесс, находящийся в системной фазе и дольше всех находящийся в ОП;
  - 5. процесс, находящийся в системной фазе;
  - v 6. процесс, находящийся в системной фазе и занимающий в ОП наибольшее место;
148. Механизм квантования времени реализует
- 1. подсистема управления файлами;
  - v 2. подсистема управления процессами;
  - 3. подсистема управления устройствами;
  - 4. ни одна из перечисленных подсистем;
149. Что из перечисленного является библиотечной функцией
- v 1. `Printf()`;
  - 2. `Open()`;
  - 3. `Creat()`;
  - 4. `Read()`;
  - 5. среди перечисленных нет ни одной библиотечной функции;
150. Если первый аргумент системного вызова `kill()` меньше нуля и не равен -1, то
- 1. ядро посыпает сигнал процессу с идентификатором, равным модулю аргумента;
  - 2. ядро посыпает сигнал всем процессам, входящим в одну группу с этим процессом;
  - v 3. ядро посыпает сигнал всем процессам, идентификатор группы которых равен этому модулю этого аргумента;
  - 4. ничего из предложенного неверно;
151. Какая из перечисленных функций возвращает дескриптор файла
- 1. `Read()`;
  - 2. `Lseek()`;
  - v 3. `Chmod()`;
  - 4. `Pipe()`;
152. В семафор не входит
- 1. значение семафора;
  - 2. идентификатор процесса, который хронологически последним работал с семафором;
  - v 3. число процессов, ожидающих увеличения семафора;
  - 4. число процессов, ожидающих нулевого значения семафора;
  - 5. число процессов, ожидающих уменьшения семафора;