**Контекст потока 31**

Содержимое памяти, к которой поток имеет доступ во время своего исполнения.

**Асинхронный ввод-вывод 499**

При синхронной записи данных в файл поток, выдавший команду записи, блокируется до тех пор, пока данные не будут записаны в файл или буфер. При асинхронной записи данных в файл, после выдачи команды на запись данных, поток не блокируется, а продолжает свою работу. Соответственно, при синхронном чтении данных из файла поток, выдавший команду чтения, блокируется до тех пор, пока данные не будут прочитаны из файла. При асинхронном чтении данных такой блокировки не происходит, и поток продолжает свое исполнение.

**Асинхронная процедура 485**

Асинхронной процедурой называется функция, которая исполняется асинхронно в контексте какого-либо потока. Для исполнения асинхронной процедуры необходимо выполнить три условия:   
- определить асинхронную процедуру;   
- определить поток, в контексте которого эта процедура будет выполняться;   
- дать разрешение на выполнение асинхронной процедуры.

**Очередь асинхронных процедур 485**

Каждый поток имеет очередь асинхронных процедур и, вызвав функцию QueueuserAPC, мы помещаем нашу асинхронную процедуру в эту очередь. Очередь асинхронных процедур работает по алгоритму FIFO (см. разд. 20.3). Фактически каждый поток имеет две очереди асинхронных процедур: одну пользовательскую, а вторую системную. В пользовательскую очередь асинхронная процедура ставится при помощи функции QueueuserAPC, а в системную очередь асинхронная процедура ставится операционной системой.

**Диспетчеризация потоков 38**

Время работы процессора делится на кванты (интервалы), которые выделяются потокам для работы. По истечении кванта времени исполнение потока прерывается и процессор назначается другому потоку. Распределением квантов времени между потоками занимается специальная программа, которая называется *менеджер потоков*.

Когда менеджер потоков переключает процессор на исполнение другого потока, он должен выполнить следующие действия:

- сохранить контекст прерываемого потока;   
- восстановить контекст запускаемого потока на момент его прерывания;   
- передать управление запускаемому потоку.

**Процедуры завершения асинхронной записи 528**

Возможен другой подход к оповещению потока о завершении асинхронной операции записи или чтения данных из файла. В этом случае операционная система не устанавливает событие, дескриптор которого задан в структуре типа overlapped, а, по завершении асинхронной операции ввода-вывода, вызывает функцию, которая называется процедурой завершения ввода-вывода {FilelOCompletionRoutine).