pthread_exit is called from the thread itself to terminate its execution (and return a result) early.

pthread_join is called from another thread (usually the thread that created it) to wait for a thread to terminate and obtain its return value. It can be called before or after the thread you're waiting for calls pthread_exit. If before, it will wait for the exit to occur. If after, it simply obtains the return value and releases the pthread_t resources.

pthread_detach can be called from either the thread itself or another thread, and indicates that you don't want the thread's return value or the ability to wait for it to finish. This is useful because otherwise, until you call pthread_join, the pthread_t value remains valid and consumes resources - at the very least, resources to store the return value and tying up one possible value of pthread_t. If you're using pthread_detach, normally you call it from either the new thread or the creating thread as soon as the new thread is created (right after pthread_create).

pthread_cancel requests that the thread teminate at the next cancellation point, and is probably safer than using pthread_kill

The pthread_cancel() function sends a cancellation request to the thread thread. Whether and when the target thread reacts to the cancellation request depends on two attributes that are under the control of that thread: its cancelability state and type. It decided by attr of thread_create.