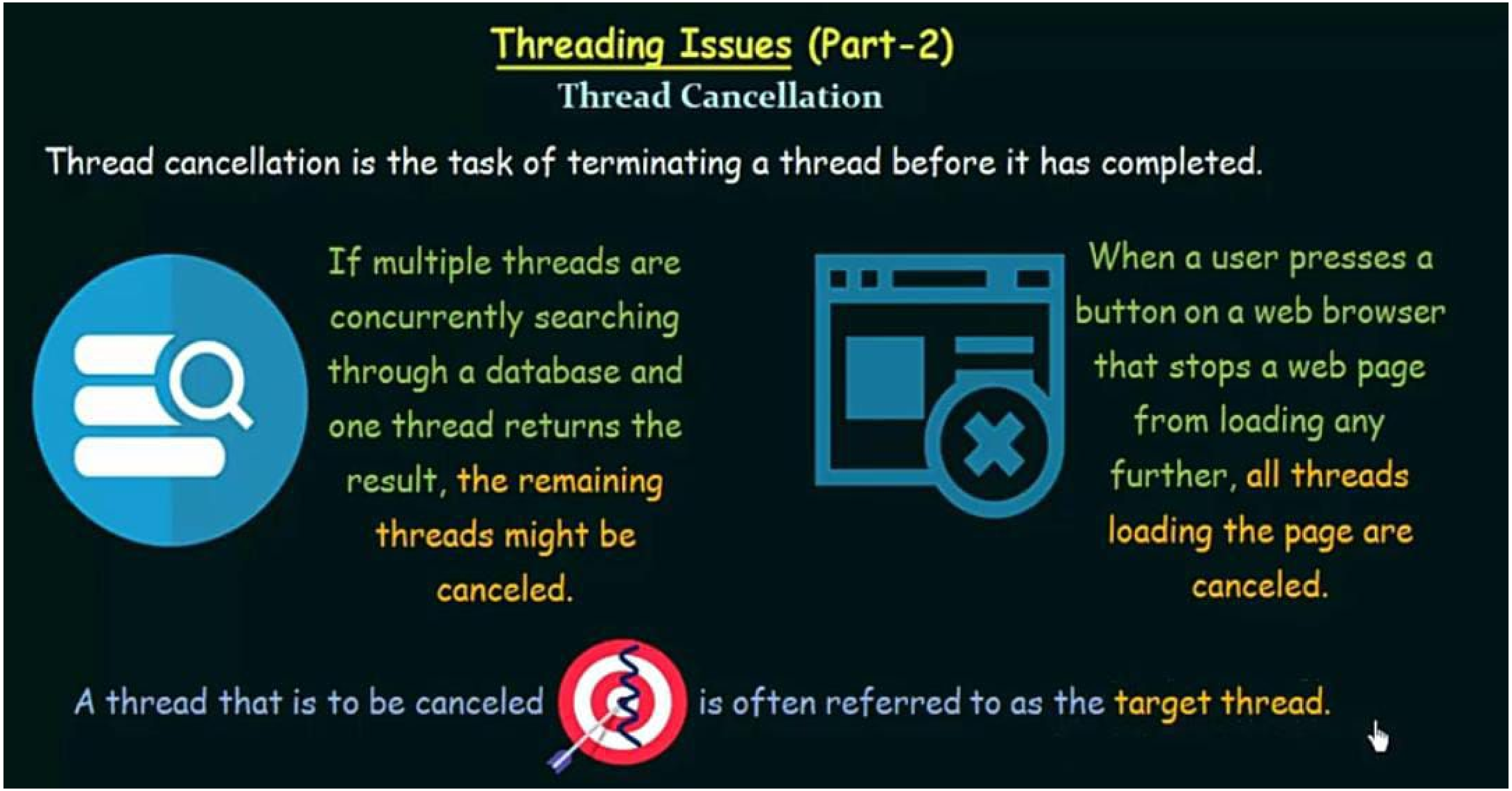


Threading Issues (Part-2)

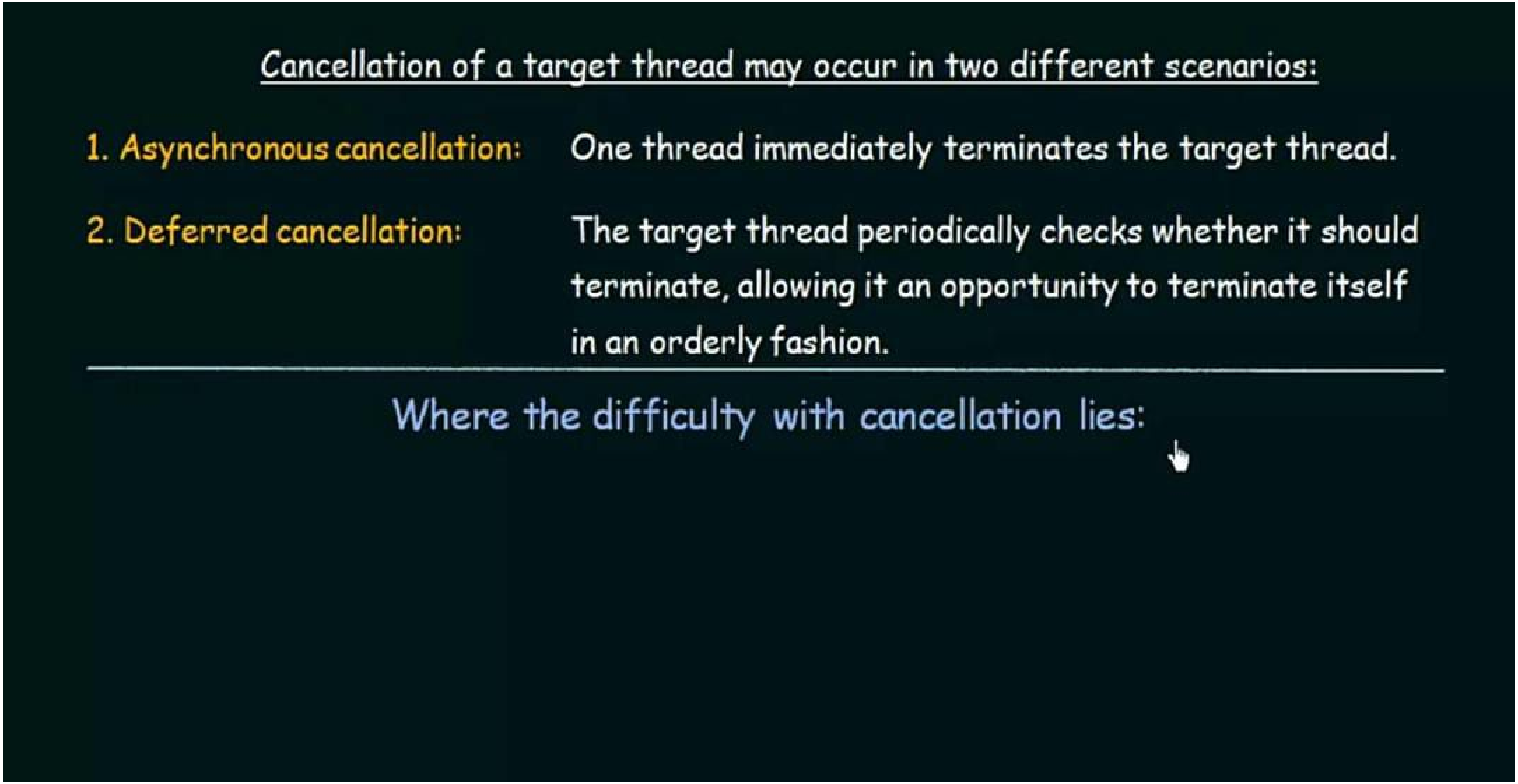
Thread Cancellation

Thread cancellation is the task of terminating a thread before it has completed.

If multiple threads are concurrently searching through a database and one thread returns the result, the remaining threads might be canceled.

When a user presses a button on a web browser that stops a web page from loading any further, all threads loading the page are canceled.

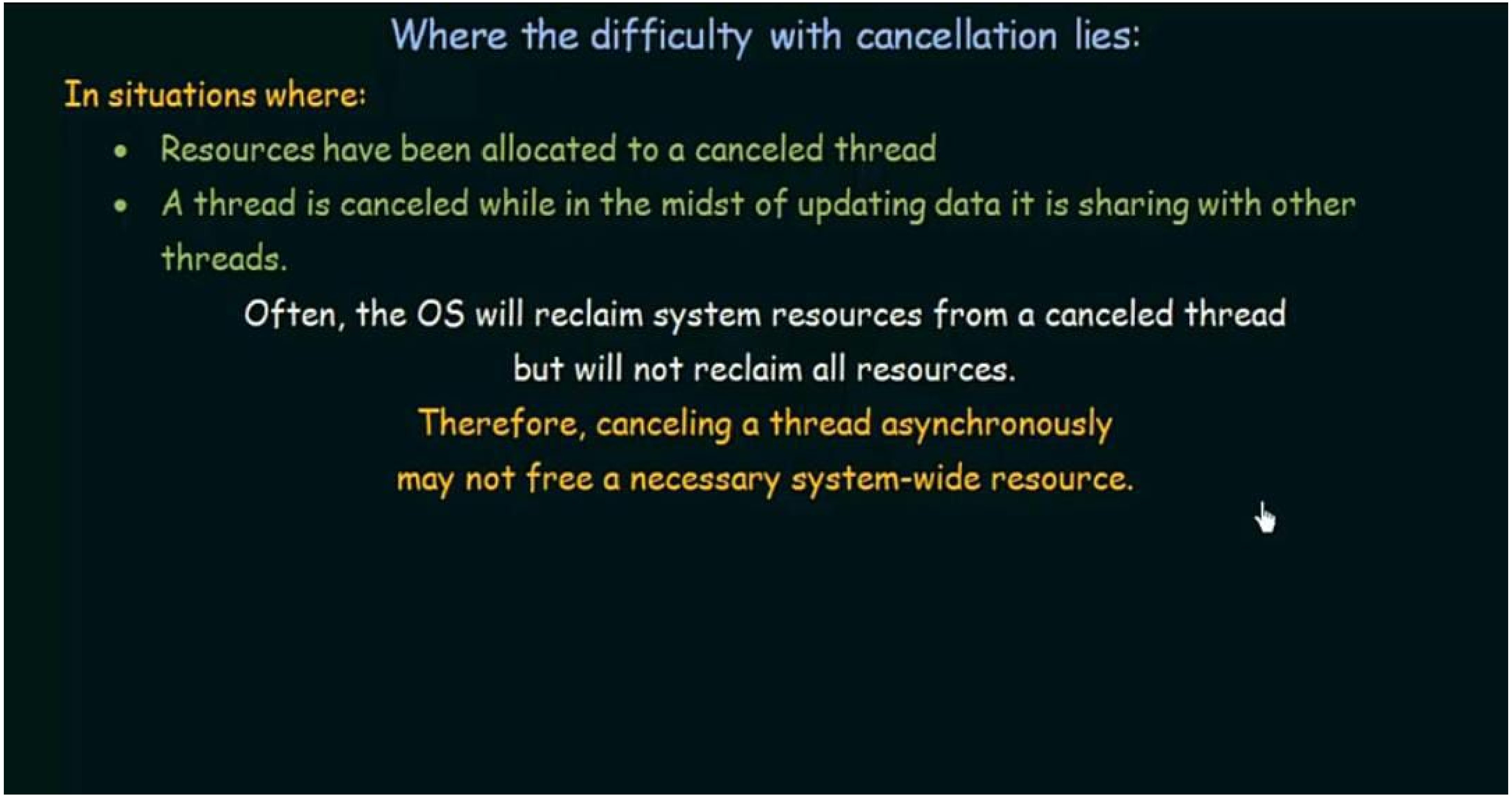
A thread that is to be canceled is often referred to as the target thread.



Cancellation of a target thread may occur in two different scenarios:

1. Asynchronous cancellation:	One thread immediately terminates the target thread.
2. Deferred cancellation:	The target thread periodically checks whether it should terminate, allowing it an opportunity to terminate itself in an orderly fashion.

Where the difficulty with cancellation lies:



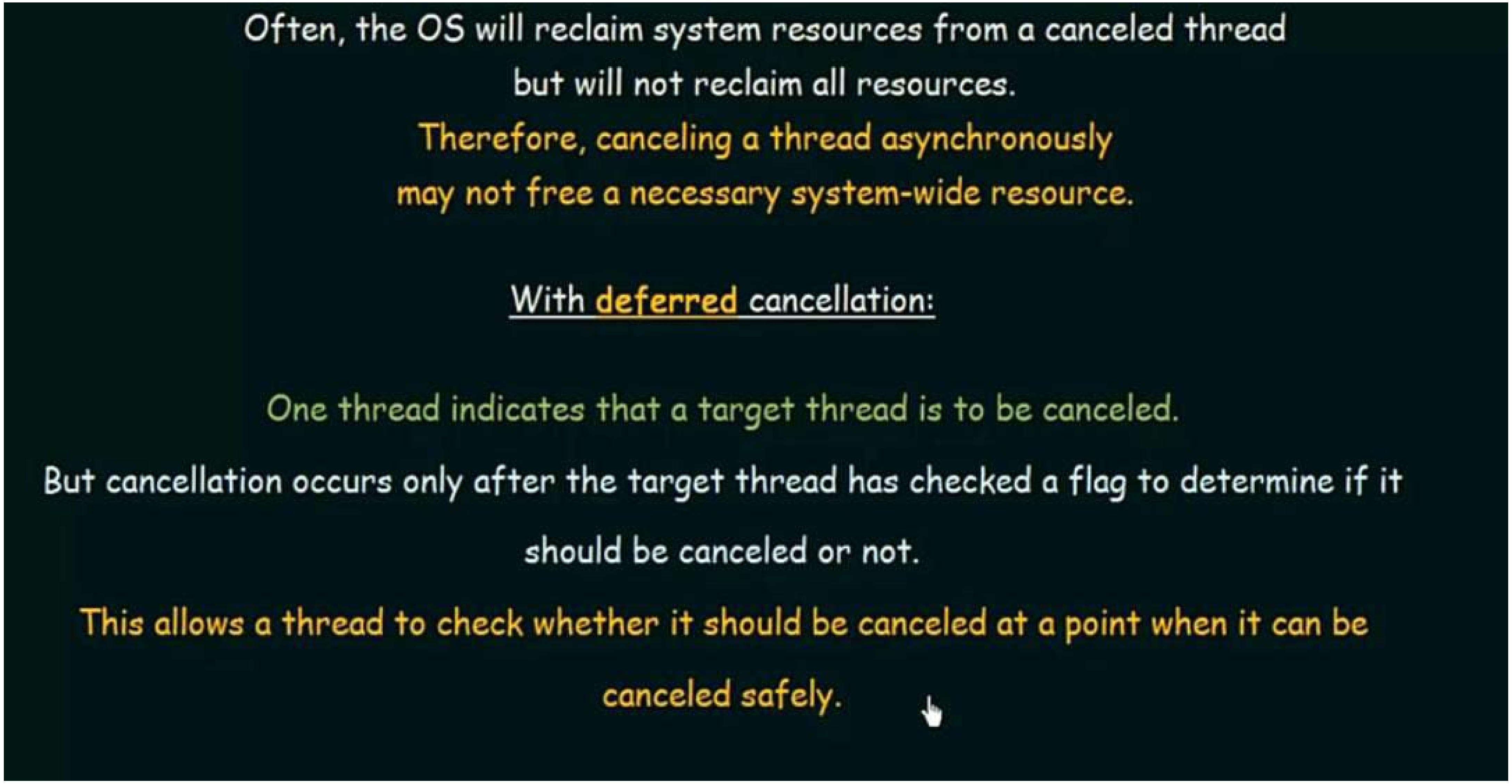
Where the difficulty with cancellation lies:

In situations where:

- Resources have been allocated to a canceled thread
- A thread is canceled while in the midst of updating data it is sharing with other threads.

Often, the OS will reclaim system resources from a canceled thread but will not reclaim all resources.

Therefore, canceling a thread asynchronously may not free a necessary system-wide resource.



Often, the OS will reclaim system resources from a canceled thread
but will not reclaim all resources.
Therefore, canceling a thread asynchronously
may not free a necessary system-wide resource.

With deferred cancellation:

One thread indicates that a target thread is to be canceled.

But cancellation occurs only after the target thread has checked a flag to determine if it should be canceled or not.

This allows a thread to check whether it should be canceled at a point when it can be canceled safely.

```
# include<stdio.h>
# include<conio.h>
 int main ( )
  {
  if (fork ( ) && fork ())
      fork ( );
    printf ("engineer");
     return 0;
     }
```

$$\begin{tikzcd}
 & P \arrow[dl] \arrow[dr] & & & \\
\circ\ C_1 \arrow[d] & & P \arrow[dl] \arrow[dr] & & \\
\text{engineers} & \circ\ C_2 & & P\ +\underline{ve} \arrow[ddl] \arrow[ddr] & \\
 & \text{engineers} & & & \\
 & & C_3 & & P \\
 & & \text{engineers} & & \text{engineers}
\end{tikzcd}$$