CancelIoEx can cancel I/O on console input, which is kind of nice

devblogs.microsoft.com/oldnewthing/20150323-00

March 23, 2015

Raymond Chen

Today's Little Program asks you to type something, but gives you only two seconds to do it. This is not interesting in and of itself, but it shows you how to cancel console I/O. There is no motivation for this exercise because Little Programs come with little to no motivation.

Okay, fine, here's the motivation.

> We have a GUI application that has a debug console. When the user exits the application, we cannot shut down cleanly because the debug console is stuck on a read from `stdin`. We want to unstick the thread gently. We don't want to use `GenerateConsoleCtrlEvent` with `CTRL_C_EVENT` because that will send the event to all processes using the same console, but we don't want other processes to be affected.

Okay, now our Little Program.

```
#include <windows.h>
#include <stdio.h> // horrors! mixing C and C++!

DWORD CALLBACK ThreadProc(void *)
{
 Sleep(2000);
 CancelIoEx(GetStdHandle(STD_INPUT_HANDLE), nullptr);
 return 0;
}

int __cdecl wmain(int, wchar_t **)
{
 DWORD scratch;
 HANDLE h = CreateThread(nullptr, 0, ThreadProc,
                         nullptr, 0, &scratch);
 if (h) {
  printf("type something\n");
  char buffer[80];
  if (fgets(buffer, 80, stdin) != nullptr) {
   printf("you typed %s", buffer);
  } else if (feof(stdin)) {
   printf("end-of-file reached\n");
  } else if (ferror(stdin)) {
   printf("error occurred\n");
  }
 }
 return 0;
}
```

If you type something within two seconds, it is reported back to you, but if you take too long, then the `CancelIoEx` cancels the console read, and you get an error back.

If you want to continue, you'll have to `clearerr(stdin)`, but if you just want to unstick the code that is performing the read (so that you can get the program to exit cleanly), then leaving `stdin` in an error state is probably better.

(If you had used `ReadFile` instead of `fgets`, the read would have failed with error code `ERROR_OPERATION_ABORTED`, as documented by `CancelIoEx`.)

Raymond Chen

Follow