

CPSC 240 Lecture

The States of a Data Stream

Background

This is an abbreviated summary off the topic of the states of a data stream. Be informed that this survey is very brief. For detailed information consult any C++ book with a chapter on streams. There you will encounter more details than will be presented here. For those who want in depth knowledge of this specific subject should visit <http://www.cplusplus.com/reference/ios/ios/good/>

Forward

The name `stdin` is a reserved word representing a pointer to the standard input stream. Usually this is the keyboard.

All streams have 4 states with each state being on (true) or off (false). The names of the 4 states are good, failed, eof, and bad.

When a program begins executing the good state is on. An attempt to read data from a stream beyond the end of data (EOF) will put the stream in good state off and failed state on.

When the failed state of a stream is on no data can pass through that stream. In the case of stream `stdin`, when `stdin` has failed state on no data can be received from the keyboard. In order to resume receiving inputs from the keyboard the states must be reset, namely: good = on and failed = off.

CNTL+D

When a user enters CNTL+D two things happen: -1 is copied to EAX and the failed bit of `stdin` is set to on (failed state = true). In the failed state the software cannot input any data – not even single characters like 'y' or 'n'. After a CNTL+D the executing program must return the stream `stdin` to a good state. The assembly block of instructions to do this is:

```
mov rax,0
mov rdi,[stdin]
call clearerr
```

Have a nice evening,
Floyd Holliday

2019-September-19 at 9:42pm

Students: Do not view this document as a technical description of the states of C++ streams. C++ textbooks will do a much better job of explaining I/O streams.