

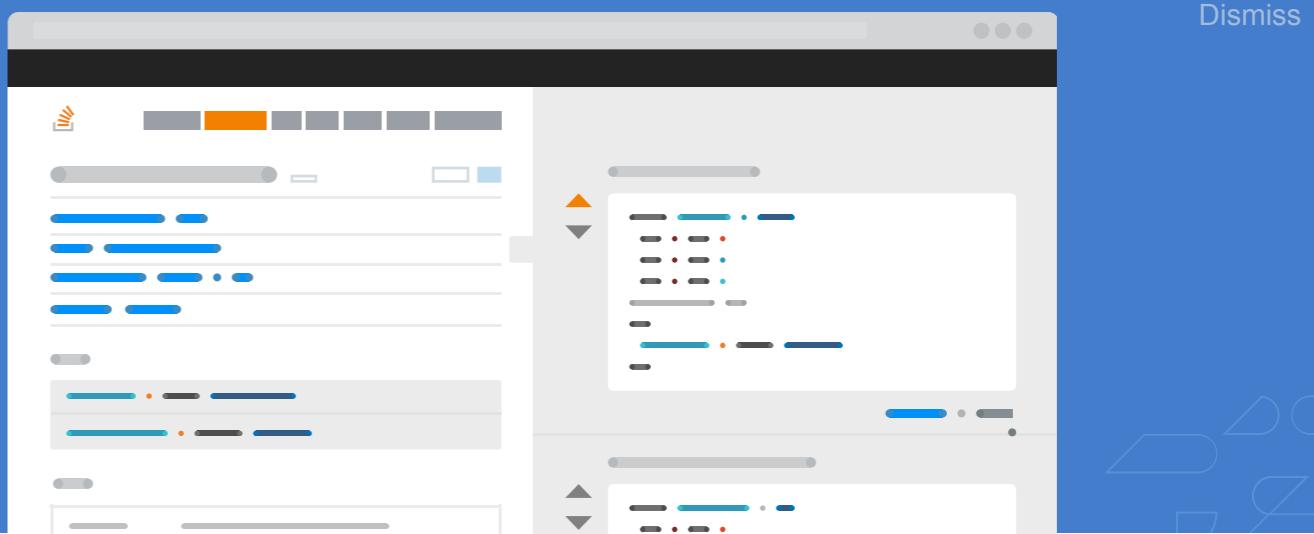
Announcing Stack Overflow Documentation

We started with Q&A. Technical documentation is next, and we need your help.

Whether you're a beginner or an experienced developer, you *can* contribute.

[Sign up and start helping →](#)

[Learn more about Documentation →](#)



Why can't we read one character at a time from System.in?



Monetize your app with over 1 million Google advertisers

[Learn more](#)



The program below prints each character written on standard in, but only after a new-line has been written (at least on my system!).

```
public class Test {  
    public static void main(String[] args) throws java.io.IOException {  
        int c;  
        while ((c = System.in.read()) != -1)  
            System.out.print((char) c);  
    }  
}
```

This prevents people from writing stuff like "Press any key to continue" and forces something like "Press enter to continue."

- What is the underlying reason for this?
- Is it a limitation of Java?
- Is this behavior system-dependent (I'm on Ubuntu)? How does it work on Mac? Windows?
- Is it dependent on the specific terminal I run the application in? (For me it behaves like this in Eclipse and in gnome-terminal)
- Is there a workaround?

[java](#) [stdin](#)

edited Apr 21 '11 at 15:21

 **Marc Mutz - mmutz**
14.7k ● 6 ● 54 ● 73

asked Oct 24 '10 at 8:27

 **aioobe**
246k ● 51 ● 550 ● 635

see my answer in [stackoverflow.com/questions/1864076/...](#) – [nagarajub](#) Apr 12 '12 at 7:16

3 Answers

What is the underlying reason for this?

Most terminals is line buffered by default, Java does not receive input until a newline.

Is it a limitation of Java?

Some ancient terminals might only have line-buffered input; though it should be possible to disable buffering in most modern terminal.

Is this behavior system-dependent (I'm on Ubuntu)? How does it work on Mac? Windows?

Yes.

Is it dependent on the specific terminal I run the application in? (For me it behaves like this

in Eclipse and in gnome-terminal)

Yes.

Is there a workaround?

There are platform specific hacks. `curse` in Linux and Unix-like platforms, and `getch()` in Windows. I'm not aware of any cross-platform way.

related: Why "Press any key to continue" is bad idea:



answered Oct 24 '10 at 8:39

 Lie Ryan
34.3k • 8 • 58 • 109

```
36 if (dev.isBored() || job.sucks()) {  
37     searchJobs({flexibleHours: true, companyCulture: 100});  
38 }  
39 A career site that's by developers, for developers.
```

 stackoverflow
JOBS

[Get started](#)

I'm on Ubuntu

Is there a workaround?

```
Runtime.getRuntime().exec("stty -icanon min 1").waitFor();
```

And after that all reads of System.in in the same process will read 1 character not waiting for EOL.

edited Jan 18 '15 at 10:36

community wiki

2 revs

hoat4

see my answer in [Equivalent function to C's "_getch\(\)" in Java?](#)

```
public static void getch() {  
  
    final JFrame frame = new JFrame();  
    synchronized (frame) {  
        frame.setUndecorated(true);  
        frame.getRootPane().setWindowDecorationStyle(JFrame.FRAME);  
        frame.addKeyListener(new KeyListener() {  
            public void keyPressed(KeyEvent e) {  
                synchronized (frame) {  
                    frame.setVisible(false);  
                    frame.dispose();  
                    frame.notify();  
                }  
            }  
  
            public void keyReleased(KeyEvent e) {  
            }  
        });  
    }  
}
```

```
    public void keyTyped(KeyEvent e) {  
    }  
});  
frame.setVisible(true);  
try {  
    frame.wait();  
} catch (InterruptedException e1) {  
}  
}  
}
```

answered Apr 12 '12 at 7:21

 **nagarajub**
751 ● 5 ● 11

2 This doesn't read from `System.in` *at all* though. – [aioobe](#) Apr 12 '12 at 8:42
