



Marco Aiello

Distributed  
Systems and  
Software Engi-  
neering group  
RuG

## Do computers *panic*?

October 20, 2014

Panic is a mental or emotional state of terror. From Greek mythology it was believed to be induced by the god Pan, hence the name. Whenever a human being or an animal would hear the shout of Pan, terror would occur as a reaction. Even if we disregard the mythological anecdote, we usually associate panic to humans or animals with developed neurological systems. But one may still wonder whether computers can panic too. And if so, what would make a computer be in a state of terror?

Let us begin by excluding a somewhat futuristic case of a machine fully capable of Artificial Intelligence. In other words, we will not ask ourselves whether or not HAL, the intelligent computer of Kubrick's 2001 a Space Odyssey movie, was in fact in panic when turning against the astronauts on board the spaceship. Let us rather focus on today's computers and their software, something like a mobile phone or a laptop. Can these machines panic? Could it happen that if I take a photograph of Bill Gates with my phone, the machine will panic?

To answer these questions, we must begin by looking at how a computer works, at least from a high-level point of view. A computer is a set of electronic components (hardware), a set of applications that are what we interact with (software) and also something sitting in the middle. What is in the middle is known as the operating system: a very special and essential software that manages the hardware offering a coherent set of high-level functionalities to the applications and, in turn, to the user. For instance, if we want to take a photograph with our mobile phone, we start the camera application, which in turn calls the operating system requesting to activate the camera hardware.

Today's operating systems are very complex. Take for instance Symbian OS, one of the most popular operating systems for mobile and smart phones to be found in Nokia, Sony-Ericsson and many other brands. It is made of over 30 million instructions and organized in several software layers. All this software is carefully designed and implemented to efficiently and effectively manage the phone, but it is not possible to foresee all states in which the phone will be and ensure that they are all correct.

But let us go back to our photograph example. Say you are walking in downtown Groningen and you see a face that looks familiar. Could that be... Yes! Bill Gates is walking around the Grote Markt looking for the Zernike complex. You don't panic, and instead you reach for your phone to take a picture, as your friends wouldn't believe you otherwise. Now, you want to save the photo you just



made. The application asks the operating system to store the photo somewhere on the phone's hardware. Normally, this will work fine, but what if there is no place left to store the photo. Should the phone panic? Not just about yet. The phone can inform the application of this unexpected circumstance. In Symbian terms, the operating system will generate a *leave*. At this point there is no reason for the application to panic. What will happen, in fact, is that the application will *trap* the leave; that is, there are some instructions ready to capture (*trap*) this exceptional circumstance and perform a compensating action. Something like asking the user to erase photos before saving the one just taken.

Though rare, it may happen that some leaves have not been anticipated by the designers of the software and that there is no trap built in to handle them. Say that now you want to send the picture to your friend using the data connection of your phone. Unfortunately, the magnetic presence of Bill affects all wireless communications around you and your phone connection fails. This will generate a leave that the photo application did not consider and therefore did not trap. What will happen to the application now? It tried to send the photo to your friend and for some unexpected reason it can't. While you sit in frustration, the application will 'panic' and, as a consequence, shut down.

In Symbian terminology, a *panic* is an exception that causes the halting of the execution. A panic can be raised by the operating system or be induced by a specific panic instruction. In fact, it can even be worse. If a panic occurs in the operating system, the device will reboot. In our example, when the camera application panics, it will abruptly stop and you will lose the photo of Bill forever.

One question remains open. Is panic a negative state to avoid? Not necessarily, in fact it often has the positive effect of making the panicking agent flee away from danger, thus saving him. Such positive effects occur also in operating systems. If a piece of software is under a malicious attack, for instance when badly formed requests are made to it, a situation known as a denial-of-service attack, another piece of software can detect it and induce panic in the attacked component. The latter will terminate execution and therefore save the phone from danger. Sometimes Pan's shouting does have positive consequences.

## References

- Jo Stichbury. Symbian OS Explained: Effective C++ Programming for Symbian OS Smartphones. John Wiley & Sons, 2008.
- <http://www.symbian.com/>
- [http://en.wikipedia.org/wiki/Kernel\\_panic](http://en.wikipedia.org/wiki/Kernel_panic)