These tutorials are a simplified introduction, and are not sufficient on their own to achieve system safety. You are responsible for the safety of your system.

Now that's what I call a dead parrot.

– John Cleese (Monty Python)
Anti-Patterns for Dependability:
- No concrete dependability goal
- Confusing reliability vs. availability
- Mission time is life of product

Can you trust your system?
- Availability: fraction of up-time
- Reliability: probability system will complete a mission
- Other properties, such as:
  - Maintainability
  - Integrity
  - Confidentiality
  - Safety
Availability

- Availability is “up time”
  \[ Availability = \frac{UpTime}{TotalTime} \]

- Limits to availability
  - Frequency of system failures
    - Redundancy can improve availability
  - Detection & repair time
    - Detect, diagnose, repair failed component, restart the system
    - Time to reconfigure to redundant standby
  - As a practical matter, 99.999% is considered “high availability”
    - 99.999% “Five nines” \(\Rightarrow\) \(~5\) minutes/year down time
    - 99.9999% “Six nines” \(\Rightarrow\) \(31.5\) seconds/year down time

- Hours Since Last System Crash:
  0 0 0 0 0 3

  99.9999% Availability Target:
  \(= 2.6\) seconds/month downtime
MS blames lowly techie for Web blackout

Takes 22 hours to fix router config error

By John Leyden 25 Jan 2001 at 11:48

Microsoft has blamed a lowly technician for a cock-up which almost completely blocked access to its Web sites for most users yesterday.

From the early hours of yesterday morning until late evening www.microsoft.com, msn.com, expedia.co.uk and msnbc.com were all unavailable. The software giant's Hotmail service was also inaccessible for many.

The problem, whose final resolution came some six hours after Microsoft promised a fix would be in place yesterday, was due to changes in Microsoft's domain name server network caused requests to access its Web sites to fail. A fix was eventually put in place when Microsoft removed the changes made to the configuration that were behind the problem.

In a statement, Microsoft admitted: "At 6:30 p.m. Tuesday (PST), a Microsoft technician made a configuration change to the routers on the edge of Microsoft's Domain Name Server network. The DNS servers are used to connect domain names with numeric IP addresses (eg. 207.46.230.219) of the various servers and networks that make up Microsoft's Web presence.

"The mistaken configuration change limited communication between DNS servers on the Internet and Microsoft's DNS servers. This limited communication caused many of Microsoft's sites to be unreachable (although they were actually still operational) to a large number of customers."

https://www.theregister.co.uk/2001/01/25/ms_blames_lowly_techie/
Reliability is based on the concept of a “mission”

- Reliability $R(t)$: probability system still working since start of mission
- A mission is $t$ continuous operating hours between diagnostics
- Constant Failure Rate $\lambda$ (failures/hr)

$$R(t) = e^{-\lambda t}$$
Serial reliability
- Even good components aren’t enough
- E.g.: $0.9 \times 0.9 \times 0.9 = 0.73$

$$R(t)_{\text{SERIAL}} = R(t)_1 \times R(t)_2 \times R(t)_3 = \prod_{i} R(t)_i$$

Parallel reliability
- Redundancy improves reliability
- E.g., three @ 0.9 $\Rightarrow$ 0.999

$$R(t)_{\text{PARALLEL}} = 1 - \left[ (1 - R(t)_1)(1 - R(t)_2)(1 - R(t)_3) \right]$$

$$R(t)_{\text{PARALLEL}} = 1 - \prod_{i} (1 - R(t)_i)$$
Reliability at MTBF $R(1/\lambda)$ is 36.8%, not 50%. Why?

What is reliability of this system for 3 hour mission?

- $\lambda_1 = 7$ per million hours
- $\lambda_2 = 200$ per million hours
- $\lambda_3 = 15000$ per million hours
- $\lambda_4 = 2$ per million hours
- $R(3)_1 = e^{-3\times7\times10^{-6}} = 0.999979$
- $R(3)_2 = e^{-3\times200\times10^{-6}} = 0.999400$
- $R(3)_3 = e^{-3\times15000\times10^{-6}} = 0.955997$
- $R(3)_4 = e^{-3\times2\times10^{-6}} = 0.999994$

- $R(3)_{PARALLEL} = 1 - [(1 - R(3)_1)(1 - R(3)_2)(1 - R(3)_3)] = 0.999 999 999 45$
- $R(3)_{TOTAL} = R(3)_{PARALLEL} R(3)_4 = 0.999 999 999 45 \times 0.999994 = 0.999994$
Other Aspects of Dependability

- **Availability**: up-time fraction
- **Reliability**: no failures
- **Safety**: no mishaps, no loss events
- **Confidentiality**: no disclosures
- **Integrity**: no corruption of state
- **Maintainability**: system can be fixed
  - E.g., “80% of failures can be fixed in 1 hour”

**Fault progression:**
- A *fault* is something that goes wrong (e.g., bit flip)
- An *error* is an activated fault (e.g., flipped bit is read and used in a calculation)
- A *failure* is when system does not provide required service (e.g., incorrect output)

A. Avizienis ; J.-C. Laprie ; B. Randell ; C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," IEEE Trans. Dependability, Jan-Mar 2004, pp. 11-33
Best Practices For Dependability

- Specify a dependability target
  - "Never fails" is unrealistic
  - Do you care about reliability or availability?

- Minimize impact of any faults
  - Fault $\rightarrow$ Error $\rightarrow$ System Failure
  - Parallel redundancy usually helps
  - Fast detection and reconfiguration

- Pitfalls:
  - Long missions without redundancy diagnosis/repair
  - Non-redundant components are weak spot $\rightarrow$ single points of failure
    - Software failures are generally neither random nor independent
  - Security matters too: attacks; outages for patches
Historical Perspective: Apollo 11 Lunar Landing

[Rocket engine burning during descent to Lunar Landing]

- **102:38:26 Armstrong:** *(With the slightest touch of urgency)* Program Alarm.
- **102:38:28 Duke:** It's looking good to us. Over.
- **102:38:30 Armstrong:** *(To Houston)* It's a 1202.
- **102:38:32 Aldrin:** 1202. *(Pause)*

[Altitude 33,500 feet.]

The 1202 program alarm is being produced by data overflow in the computer. It is not an alarm that they had seen during simulations but, as Neil [Armstrong] explained during a post-flight press conference “In simulations we have a large number of failures and we are usually spring-loaded to the abort position. And in this case in the real flight, we are spring-loaded to the land position.”

In Houston, Steve Bales, the control room's expert in the LM guidance systems, has determined that the landing will not be jeopardized by the overflow. The overflow consists of an unexpected flow of data concerning radar pointing. The computer has been programmed to recognize this data as being of secondary importance and will ignore it while it does more important computations.
Video of Apollo 11 Landing

- At the time all we had was audio – no live TV during the landing
  - 11 min. clip; HBO mini-series compressed events; overall accurate

- Things to note:
  - Collins is in the command module
  - Armstrong & Aldrin in Eagle ➔ Lunar Lander
  - 1201 & 1202 alarms light up the “abort mission” warning light
    - Computer/human interface was just a bunch of digits on a display panel
    - Total of five of these alarms (three shown in the HBO version)
  - At zero seconds of fuel remaining they’re supposed to abort
    - Jettison lower half/landing stage and return to orbit
    - Q: for Apollo 11, how many seconds of fuel were left when they landed?