Computer Reliability

Introduction

• Computer systems are sometimes unreliable
  – Erroneous information in databases
  – Misinterpretation of database information
  – Malfunction of embedded systems
• Effects of computer errors
  – Inconvenience
  – Bad business decisions
  – Fatalities
• What are the types of errors we see?
• What can we do about them?
Data-Entry or Data-Retrieval Errors

- A computerized system may fail because wrong data entered into it.
- A computerized system may fail because people incorrectly interpret data they retrieve.

Malicious Data Entry...? vs Elegant Failure?

---

Hi, this is your son's school. We're having some computer trouble.

Oh, dear — did he break something? In a way —

Did you really name your son Robert?); DROP TABLE Students;--?

Oh, yes. Little Bobby tables, we call him.

Well, we've lost this year's student records. I hope you're happy.

And I hope you've learned to sanitize your database inputs.
**When software fails**

---

**Genesis of the Therac-25**

- AECL and CGR built Therac-6 and Therac-20
- Therac-25 built by AECL
  - PDP-11 an integral part of system
  - Hardware safety features replaced with software
  - Reused code from Therac-6 and Therac-20
- First Therac-25 shipped in 1983
  - Patient in one room
  - Technician in adjoining room
Software Errors

• Race condition: order in which two or more concurrent tasks access a shared variable can affect program’s behavior
• Two race conditions in Therac-25 software
  – Command screen editing
  – Movement of electron beam gun

Moral Responsibility of the Therac-25 Team

• Conditions for moral responsibility
  – Causal condition: actions (or inactions) caused the harm
  – Mental condition
    • Actions (or inactions) intended or willed -OR-
    • Moral agent is careless, reckless, or negligent

• Therac-25 team morally responsible
  – They constructed the device that caused the harm
  – They were negligent
Shrinkwrap Warranties

• Some say you accept software “as is”
• Some offer 90-day replacement or money-back guarantee
• None accept liability for harm caused by use of software

Moral Responsibility of Software Manufacturers

• If vendors were responsible for harmful consequences of defects
  – Companies would test software more
  – They would purchase liability insurance
  – Software would cost more
  – Start-ups would be affected more than big companies
  – Less innovation in software industry
  – Software would be more reliable

• Making vendors responsible for harmful consequences of defects may be wrong
• Consumers should not have to pay for bug fixes
Software Quality

• Standish Group tracks IT projects
• Situation in 1994
  – 1/3 projects cancelled before completion
  – 1/2 projects had time and/or cost overruns
  – 1/6 projects completed on time / on budget

• Situation in 2006
  – 1/6 projects cancelled
  – 1/2 projects had time and/or cost overruns
  – 1/3 projects completed on time / on budget

Bane of quality: Feature creep

Pentagon Wars

• https://www.youtube.com/watch?v=aXQ2lO3ieBA&list=PLCxhrvRMA5YR01vAdgFFNj4fhj850MomZ&index=4

Hitler’s take:

• https://www.youtube.com/watch?v=n35pbmbcR2s
Death of a librarian –
June 2005

• Wolf Djupedal is a librarian at a new national Norwegian library
• He has catalogued an important new collection – 14,000 books & magazines
• Does as he is supposed to, backs up & encrypts data for security purposes
• Dies without telling anyone password

• Incredibly common problem
• Lessons?

More: http://www.computerworld.com/securitytopics/security/encryption/story/0,10801,71721,00.html

Bad passwords

• In 2009 RockYou.com accidentally released 32 million passwords
• Most people are very bad at picking passwords unless coached

<table>
<thead>
<tr>
<th>Password Popularity – Top 20</th>
<th>Rank</th>
<th>Password</th>
<th>Number of Users with Password (absolute)</th>
<th>Rank</th>
<th>Password</th>
<th>Number of Users with Password (absolute)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>123456</td>
<td>290731</td>
<td>11</td>
<td>Nicole</td>
<td>17168</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12345</td>
<td>79078</td>
<td>12</td>
<td>Daniel</td>
<td>16409</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>123456789</td>
<td>76790</td>
<td>13</td>
<td>babygirl</td>
<td>16094</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Password</td>
<td>61958</td>
<td>14</td>
<td>monkey</td>
<td>15294</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>iloveyou</td>
<td>51622</td>
<td>15</td>
<td>Jessica</td>
<td>15162</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>princess</td>
<td>35231</td>
<td>16</td>
<td>Lovely</td>
<td>14950</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>rockyou</td>
<td>22588</td>
<td>17</td>
<td>michael</td>
<td>14898</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1234567</td>
<td>21726</td>
<td>18</td>
<td>Ashley</td>
<td>14329</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>12345678</td>
<td>20553</td>
<td>19</td>
<td>654321</td>
<td>13984</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>abc123</td>
<td>17542</td>
<td>20</td>
<td>Qwerty</td>
<td>13856</td>
</tr>
</tbody>
</table>
Ariane 5 Flight 501 – June 4 1996

- Maiden flight of new launch vehicle, much larger and powerful than old Ariane 4.

- Cost and reliability a factor, a lot of technology was re-used from Ariane 4
  - Less costly than developing new
  - Was already tested and proven
  - In most cases the old technology was good enough

Result: [http://www.youtube.com/watch?v=IONcgYzVFlg](http://www.youtube.com/watch?v=IONcgYzVFlg)

Ariane 5 Flight 501 – Post-mortem

• Often dubbed the worlds most expensive software bug ($370 millions)
• Unhandled exception in data conversion (64-bit Float to 16 bit signed Integer) led inertial guidance computer to crash

• What went wrong?

Mars climate orbiter
Sept 23 1999

• Designed to study weather on Mars, serve as relay for all other Mars probes
• Reached Mars properly, started descent into orbit
  – Was meant to orbit at 140-150km height
  – Attempted to enter orbit at 57km
  – Burned on entry as result

• Total cost of both Orbiter and Lander: $327 million

• More: http://en.wikipedia.org/wiki/Mars_Climate_Orbiter
Mars climate orbiter
Sept 23 1999

What went wrong?
– Metric vs. Imperial

• NASA specified thrust in Newtons (Metric), contractor assumed Pound Force (Imperial)
  – 1 pound force = 4.45 Newtons
  – On entry, thrusters fired much longer than needed

• Software had been adapted from Mars Climate Orbiter, without proper testing before launch

Airbus 320

First delivered in 1988, the A320 pioneered the use of digital fly-by-wire flight control systems in a commercial aircraft

Designed to help avoid pilot error – main cause of accidents

Included safety controls for reverse thrust:
• To ensure that the thrust-reverse system and the spoilers are only activated in a landing situation, all of the following have to be true for the software to deploy these systems:

• there must be weight of over 12 tons on each landing gear strut
• the wheels of the plane must be turning faster than 133 km/h
• the thrust levers must be in the idle (or reverse thrust) position
Lufthansa Flight 2904 – Sept 14 1993

- Lufthansa flight from Frankfurt into Warsaw on an Airbus 320-200
- Reports of some wind shear and rain, so pilots follow textbook procedure
  - Slightly higher landing speed
  - Come in slightly sideways

Brakes fail to deploy until too late, runs out of runway, copilot and 1 passenger die


---

Lufthansa Flight 2904 – Post Mortem

What went wrong?

Requirements:
- there must be weight of over 12 tons on each landing gear strut
- the wheels of the plane must be turning faster than 133 km/h
- the thrust levers must be in the idle (or reverse thrust) position
• Work started in May 2001
  – Expected to ship late 2003
  – Based on XP code
  – Meant as minor upgrade between XP and Blackcomb

• Early problems
  – Project started to absorb many of the goals of Blackcomb
  – Many core developers pulled away to write patches for XP and 2003

“Microsoft co-president Jim Allchin, who had overall responsibility for the development and delivery of Windows, explained how development of Longhorn was “crashing into the ground” due in large part to the haphazard methods by which features were introduced and integrated into the core of the operating system, without a clear focus on an end-product.”

• August 2004 reset
  – Original plans abandoned, restart project based on Windows 2003 SP1 code-base
  – Many Blackcomb features pulled out
    • WinFS
  – New development methodologies
    • Security Development Lifecycle
    • Many complaints, including from Billg

• Final launch January 30, 2007

• $6 Billion(?) price tag