ECE 471 – Embedded Systems
Lecture 28

Vince Weaver

http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

10 November 2021

Announcements

- If you need any parts for your project, let me know
- Office hours (1pm-2pm) cancelled today so I can attend a MCECIS architecture meeting. If you need anything stop by later or e-mail me.

Homework 8 – Code

- Error checking. Exit if cannot open. If you don't, can segfault if try to fscanf a NULL FILE*
- Returning -1 on error might be bad idea
- What to report on error? What's an invalid temperature? Not just unlikely? (Below Absolute zero)
- If using streams (FILE *fff), on fopen() error it returns NULL, not -1.
- Be sure to close files, otherwise leak file descriptors Be careful if multiple exit points, must close at all (goto)

- Be careful with your 9/5 Fahrenheit conversion!
- Finding a file using C. `opendir()` `readdir()`, horrible interface

  Bit of a tangent on the downsides of the `readdir()` interface

HW#8 – Questions

- Why need Vdd? To provide enough current for this particular chip needs extra current if you want parasite mode.

  You can try without Vdd but you will always read out 85C.

  Manual suggests MOSFET, but apparently it's possible on Pi if use 4.7k resistor as well as "strong-pullup=y" kernel command line option.
- Because of distance, 1-wire

- shell script
  - `#!/bin/sh` should be first line (magic number)
  - Trouble if edit on windows, why (linefeed vs carriage return)

    shebang description
  - Making executable with chmod
  - Default shell, can put other things there, like python or perl, etc, even ARM emulator
  - sh vs bash

Spacecraft

- Mariner 1 (1962) – rocket off course due to mis-transcribed specification into FORTRAN, missing overbar
- Apollo 11 (1969) – landing on moon.
  - 36k ROM (rope), 2k RAM, 70lbs, 55W, 5600 3-input NOR
  - Processor normally loaded with 85% load. DELTAH program run which take 10%. But buggy radar device was stealing 13% even though in standby mode.

  - Multiple 1202 overload alarms
  - Mini real-time OS with priority killed low-priority tasks so things still worked.
- Ariane 5 Flight 501 (1996) – famous. $370 million.
  - Old code copied from Ariane 4. Horizontal acceleration
  - Could not trigger on Ariane 4 (accel never that large)
  - Could trigger on more powerful Ariane 5
  - Conversion from 64-bit float to 16-bit signed int overflowed. Trap
  - Primary guidance computer crashed
  - Secondary computer, but ran same code, crashed

  - Sent debug messages after crash, autopilot read those as velocity data
  - Destructed 37s after launch
  - Written in ADA
- NASA Mars Polar Lander (1999)
  - likely mistook turbulence vibrations for landing and shut off engine 40m above surface
- NASA Mars Climate Orbiter
  - ground software using lbf (pound/foot) units, craft expecting Newtons
- NASA Mars Spirit rover (2004)

  - temporarily disabled due to too many files on flash drive
  - Constantly rebooting
  - Radio could understand some commands directly, could reboot with flash disabled.
  - Fixed when deleted some unneeded files.
  - Eventually reformat.
  - Issue is 90 day design period, lasted years (until 2010)
- Phobos-Grunt (2012)
  - Bit flip in memory caused it to crash before firing rockets to Mars

    - Entered safe mode waiting for command
    - Antennas not deployed until after rocket firing
    - Could not receive command to leave safe mode.
- ExoMars Schiaparelli Lander (2016)
    - Bad data to inertial measurement unit for 1 second
    - thought this meant it was below ground level, released parachute when still 3.7km up.
    - Had valid data from radar
- Boeing Starliner OTF-1 flight issues (2019-2021)
    - Lack of full-stack integration testing meant the capsule thought it was 11 hours further in mission than it was,

firing engines wildly and using up most propelant

- Last-minute firmware update saved the landing
- Earlier problem with improperly packed parachute
- Next try in 2021 last-minute abort due to valves rusting shut

Medical Example

- Therac-25 radiation treatment machine, 1985-1987
- 6 accidents, patients given 100x dose. Three died High power beam activated w/o spreader too. **Older machines had hardware interlock**, this one in software. Race condition. If 8-bit counter overflow just as entering manual over-ride, it would happen.
- Triggering the bug
  - To trigger, had to press X (mistake), up (to correct), E (to set proper) then "Enter" all within 8 seconds.

This was considered an improbable series of keypresses.

- This missed during testing as it took a while for operators to get used to using machines enough to type that fast.
- Used increment rather than move to set flag, this meant sometimes it wrapped from 255 to 0, disabling safety checks
- Written in Assembly Language

Things that went wrong with design

- Software not independently reviewed
- No reliability modeling or risk management

- Something wrong: Printed “MALFUNCTION” and error number 1 to 64 which was not documented in manual. Press P to clear.
- Operators not believe complaints from patients.
- The setup was not tested until after it was installed at hospital.
- cut-and-pasted software from earlier model that had hardware interlocks
- Concurrent (parallel) operation with race conditions

Another Medical Example

- Devices like pacemakers, how does a doctor reprogram them?
- Are they password protected?

Financial

- Knight Capital. Upgrade 7 of 8 machines, missed last. Re-used a flag definition with new software. Caused massive selloff, $440 million

Power

- 2003 Blackout
  - Power plant fail. Cause more current down transmission lines in Ohio. Heat, expand, touch tree, short out.
  - Race condition in Unix XA/21 management system, so alarms not go off
  - Eventually primary system fail as too many alarms queue up
  - Backup server also fail

- During failure, screens take 59s (instead of 1s) to update
- Blackout of most of NY and a lot of north east.

Example of Good Design – Space Shuttle Computer

- https://www.nasa.gov/mission_pages/shuttle/flyo
  flyfeature_shuttlecomputers.html
- Issues normal embedded systems don't have: Vibration at liftoff, Radiation in Space
- If computer stopped for more than 120ms, shuttle could crash
- "Modern" update in 1991: 1MB Ram, 1.4MIPS. Earlier was 416k and 1/3 as fast and twice as big

- Change to code, 9 months testing in simulator, 6 months more extensive testing
- 24 years w/o in-orbit SW problem needing patches
- 12 year stretch only 3 SW bugs found
- 400k lines of code
- HAL/S high-order assembly language (high-level language similar to PL/I)
- PASS software – runs tasks. Too big to fit in memory at once
- BFS – backup flight software. Bare minimum to takeoff, stay in orbit, safely land, fits in memory, monitors pASS

during takeoff/landing Written by completely different team.

- 28 months to develop new version
- IBM
- Extensive verification. One internal pass, one external
- 4 computers running PASS, one running BFS
- Single failure mission can continue; still land with two failures
- 4 computers in lock-step, vote, defective one kicked out