



An F-16 fighter plane on the northern hemisphere. Why the northern hemisphere, you ask?



Because **this** is what an F-16 on the southern hemisphere would look like. (BTW, interesting effect if you drop a bomb :-)

From risks.digest, volume 3, issue 44:

F-16 Landing Gear



From risks.digest, volume 3, issue 44: o One of the first things the Air Force test pilots tried on an early F-16 was to tell the computer to raise the landing gear while standing still on the runway. Guess what happened? Scratch one F-16. (my friend



Retrieved by a technician from the Harvard Mark II machine on September 9, 1947.

Now on display at the Smithsonian, Washington





Facts on Debugging

- Software bugs cost ~60 bln US\$/yr in US
- Improvements could reduce cost by 30%
- Validation (including debugging) can easily take up to 50-75% of the development time
- When debugging, some people are three times as efficient than others

8

A Sample Program

\$ sample 9 8 7
Output: 7 8 9

\$ sample 11 14
Output: 0 11







From Defect to Failure

- 1. The programmer creates a *defect* an error in the code.
- 2. When executed, the defect creates an *infection* an error in the state.
- 3. The infection propagates.
- 4. The infection causes a failure.

This infection chain must be traced back – and broken.





The Curse of Testing

- Not every defect causes a failure!
- Testing can only show the presence of errors – not their absence. (Dijkstra 1972)













The defect must be searched in _*space_* and _*time_*















Search in Space		
variables		
t		
22	22	





State of the GNU
compiler (GCC)
42991 vertices
44290 edges - and 1 is
wrong :-)
An actual GCC
execution has millions
of these states.



```
int *a;
int i;
```

```
a = (int *)malloc((argc - 1) * sizeof(int));
for (i = 0; i < argc - 1; i++)
    a[i] = atoi(argv[i + 1]);</pre>
```

```
shell_sort(a, argc);
```

```
printf("Output: ");
for (i = 0; i < argc - 1; i++)
    printf("%d ", a[i]);
printf("\n");
```

free(a);

```
return 0;
```





00
20



FIXME: argv[0] should be "sample", not "11"



















Concepts

- ★ A failure comes to be in three stages:
 - I. The programmer creates a defect
 - 2. The defect causes an infection
 - 3. The infection causes a *failure* an externally visible error.
- ★ Not every defect results in an infection, and not every infection results in a failure.

37



Concepts (3)

- ★ A variety of tools and techniques is available to *automate debugging*:
 - Program Slicing
 - Observing & Watching State
 - Asserting Invariants
 - Detecting Anomalies
 - Isolating Cause-Effect Chains

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit		
http://creativecommons.org/licenses/by/1.0		
or send a letter to Creative Commons, 559 Abbott Way, Stanford, California 94305, USA.	40	
40	40	