

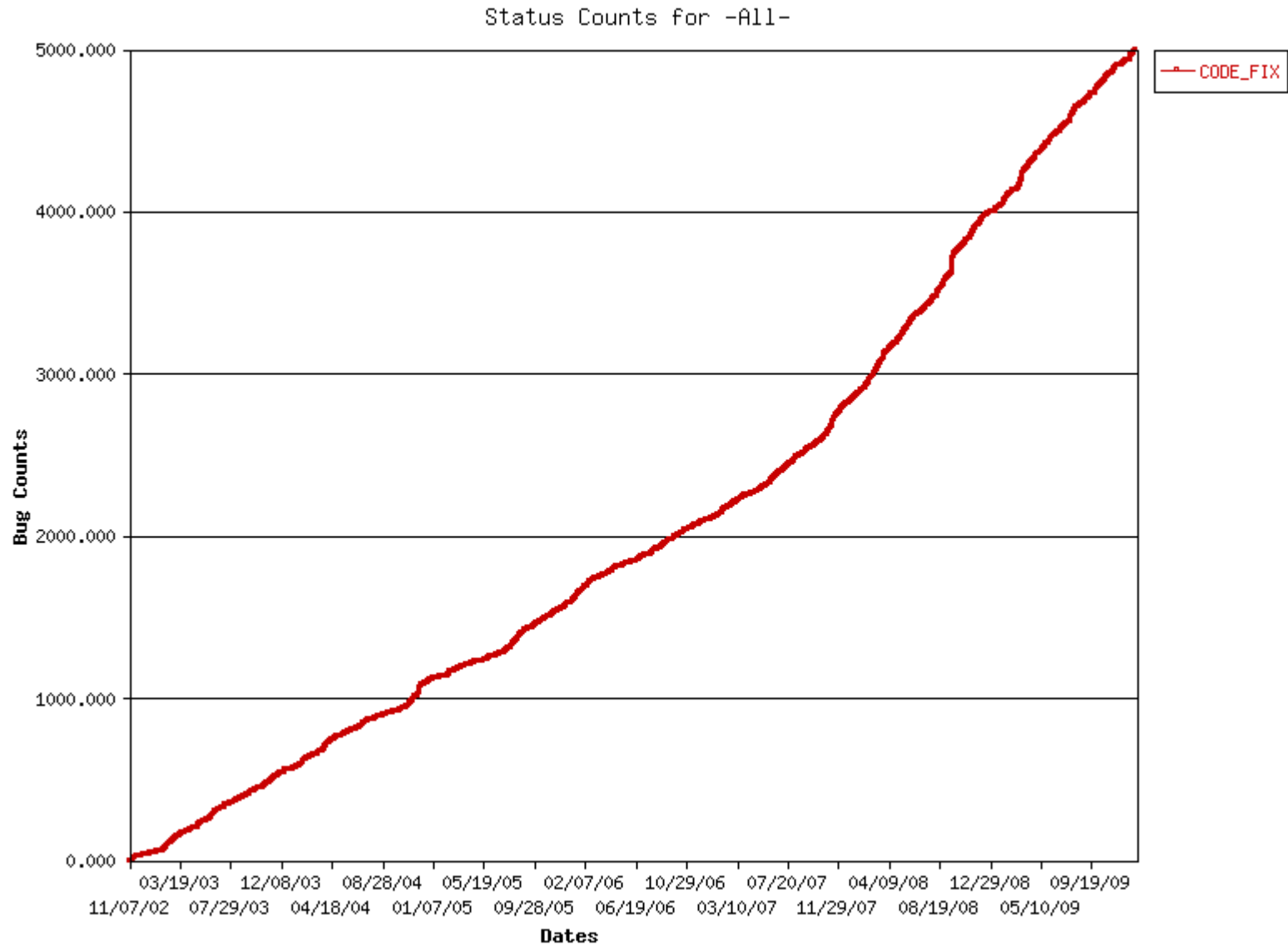
# **L8: Virtual machines**

Nickolai Zeldovich  
6.033 Spring 2012

# Kernel complexity

- 1975: Unix v6
  - 10,500 total lines of kernel code
- 2012: Linux 3.2
  - 300,000 lines: header files (data structures, APIs)
  - 490,000 lines: networking
  - 530,000 lines: sound
  - 700,000 lines: support for 60+ file systems
  - 1,880,000 lines: support for 25+ CPU architectures
  - 5,620,000 lines: drivers
  
  - 9,930,000 total lines of code

# Linux (2002-2009): avg of 2+ bugs fixed every day





```

set_ptp(guest_pt):
  for gva in 0 .. 220:
    if guest_pt[gva] & PTE_P:
      gpa = guest_pt[gva] >> 12
      hpa = host_pt[gpa] >> 12
      shadow_pt[gva] = (hpa << 12) | PTE_P
    else:
      shadow_pt[gva] = 0
  PTP = shadow_pt

```

```
set_ptp_exception(guest_pt):
    acquire(t_lock)
    id = cpus[CPU].thread
    if threads[id].vm_k:
        set_ptp(guest_pt)
    else:
        ## deliver exception to guest VM
    release(t_lock)
```



```
set_ptp(guest_pt, kmode):
```

```
  for gva in 0 .. 220:
```

```
    if guest_pt[gva] & PTE_P and  
      (kmode or guest_pt[gva] & PTE_U):
```

```
      gpa = guest_pt[gva] >> 12
```

```
      hpa = host_pt[gpa] >> 12
```

```
      shadow_pt[gva] = (hpa << 12) | PTE_P | PTE_U
```

```
    else:
```

```
      shadow_pt[gva] = 0
```

```
  PTP = shadow_pt
```

# Summary

- Monolithic OS kernels are complex, error-prone
- Microkernels
  - Enforce OS modularity with client/server (designing modular OS services is hard work)
- Virtual machines
  - Multiplex hardware between multiple OSes (does not enforce modularity within kernel)