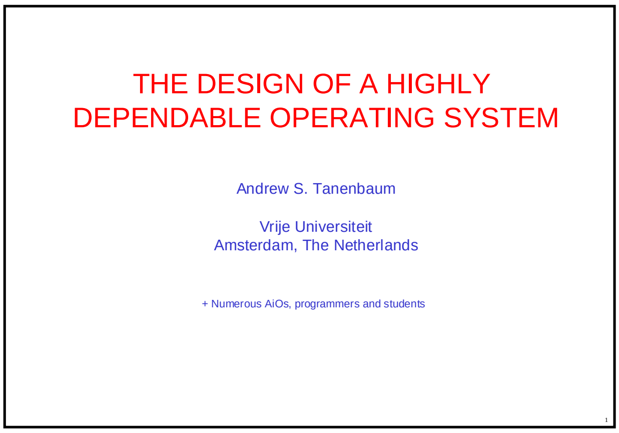
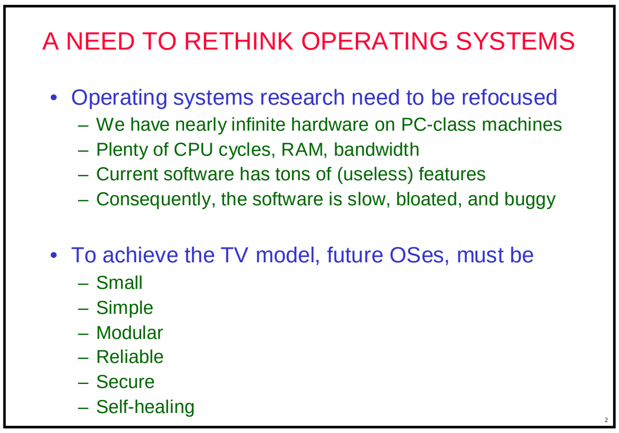


THE DESIGN OF A HIGHLY DEPENDABLE OPERATING SYSTEM

Andrew S. Tanenbaum

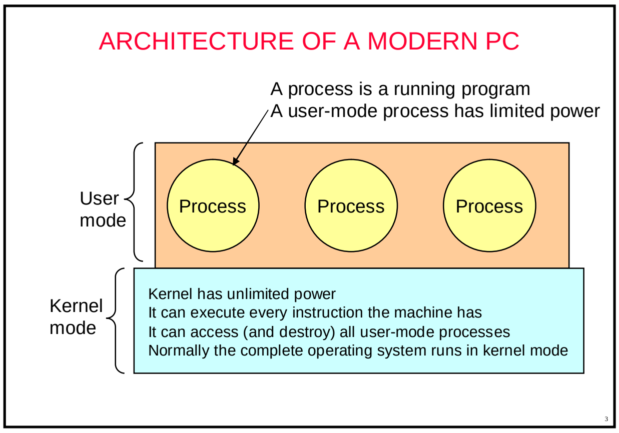
Vrije Universiteit
Amsterdam, The Netherlands

+ Numerous AiOs, programmers and students



A NEED TO RETHINK OPERATING SYSTEMS

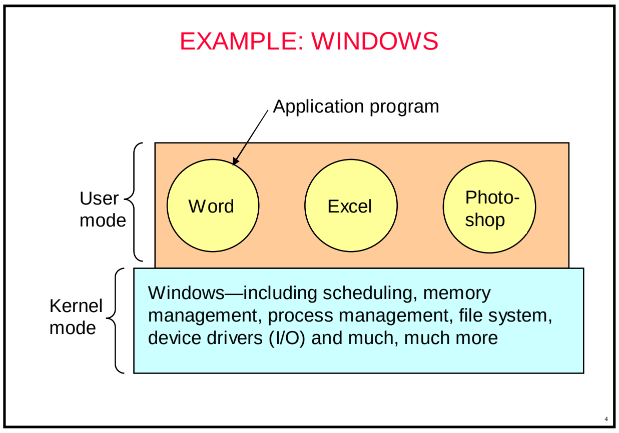
- Operating systems research need to be refocused
  - We have nearly infinite hardware on PC-class machines
  - Plenty of CPU cycles, RAM, bandwidth
  - Current software has tons of (useless) features
  - Consequently, the software is slow, bloated, and buggy
- To achieve the TV model, future OSes, must be
  - Small
  - Simple
  - Modular
  - Reliable
  - Secure
  - Self-healing



ARCHITECTURE OF A MODERN PC

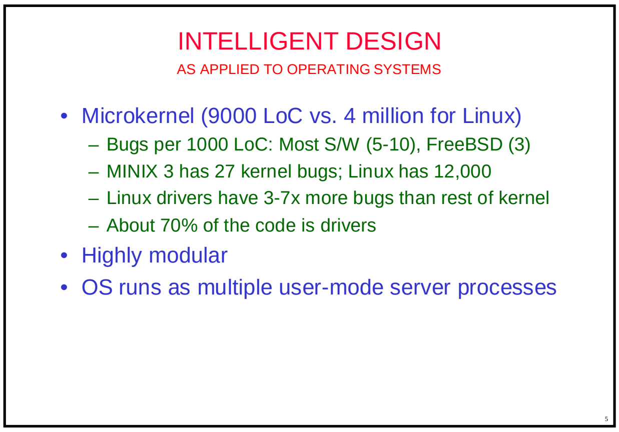
A process is a running program
A user-mode process has limited power

Kernel has unlimited power
It can execute every instruction the machine has
It can access (and destroy) all user-mode processes
Normally the complete operating system runs in kernel mode



EXAMPLE: WINDOWS

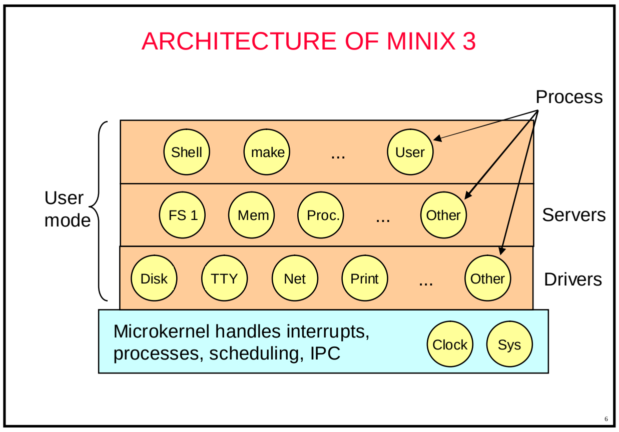
Windows—including scheduling, memory management, process management, file system, device drivers (I/O) and much, much more



INTELLIGENT DESIGN

AS APPLIED TO OPERATING SYSTEMS

- Microkernel (9000 LoC vs. 4 million for Linux)
  - Bugs per 1000 LoC: Most S/W (5-10), FreeBSD (3)
  - MINIX 3 has 27 kernel bugs; Linux has 12,000
  - Linux drivers have 3-7x more bugs than rest of kernel
  - About 70% of the code is drivers
- Highly modular
- OS runs as multiple user-mode server processes



ARCHITECTURE OF MINIX 3

Microkernel handles interrupts, processes, scheduling, IPC

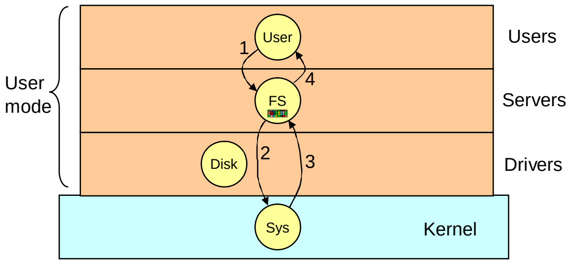
USER-MODE DEVICE DRIVERS

- Each runs as a user-mode process
- No superuser privileges
- Protected by the MMU
- Do not have access to I/O ports, privileged instrs

USER-MODE SERVERS

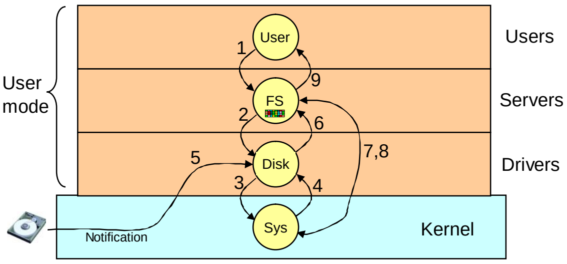
- File server
- Process manager
- Virtual memory manager
- Data store
- Information server
- Network server
- X server
- Reincarnation server

FILE SERVER (1)



File access when the block is in the FS cache

FILE SERVER (2)



File access when the block is NOT in the FS cache

DATA STORE

- Small, local name server
- Used to map server name to end point
- Could be used for recoverable drivers

INFORMATION SERVER

- Used for debug dumps

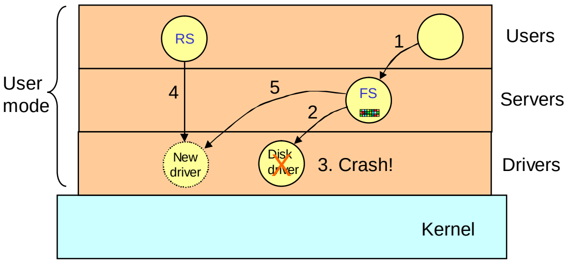
NETWORK SERVER

- Contains full TCP/IP stack in user space
- No networking code in the kernel

REINCARNATION SERVER

- Parent of all the drivers and servers
- When a driver or server dies, RS collects it
- RS checks a table for action to take e.g., restart it
- RS also pings drivers and servers frequently

DISK DRIVER RECOVERY



System is self healing

CRASHES OF OTHER DRIVERS

- Ethernet - just restart it (TCP recovers)
- Printer - line printer daemon restarts print job
- Audio - Replay the song
- etc.

KERNEL RELIABILITY/SECURITY

- Fewer LoC means fewer kernel bugs
- Small kernel means reduced TCB
- NO foreign code (e.g., drivers) in the kernel
- Static data structures (no malloc in kernel)
- Moving bugs to user space reduces their power

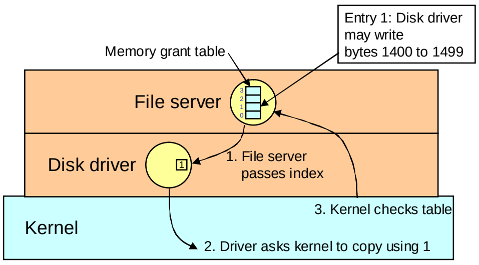
IPC RELIABILITY/SECURITY

- Fixed-length messages (no buffer overruns)
- Rendezvous system is simple
  - No lost messages
  - No buffer management
- Interrupts and messages are unified

DRIVER RELIABILITY/SECURITY

- Untrusted code: heavily isolated
- Bugs, viruses cannot spread to other modules
- Cannot touch kernel data structures
- Bad pointers crash only one driver; recoverable
- Infinite loops detected and driver restarted
- Restricted power to do damage (not superuser)

MEMORY GRANTS



Entry 1: Disk driver
may write
bytes 1400 to 1499

Memory grant table

File server

Disk driver

1. File server
passes index

3. Kernel checks table

Kernel

2. Driver asks kernel to copy using 1

OTHER ADVANTAGES OF USER DRIVERS

- Short development cycle
- Normal programming model
- No down time for crash and reboot
- Easy debugging
- Good flexibility

FAULT INJECTION

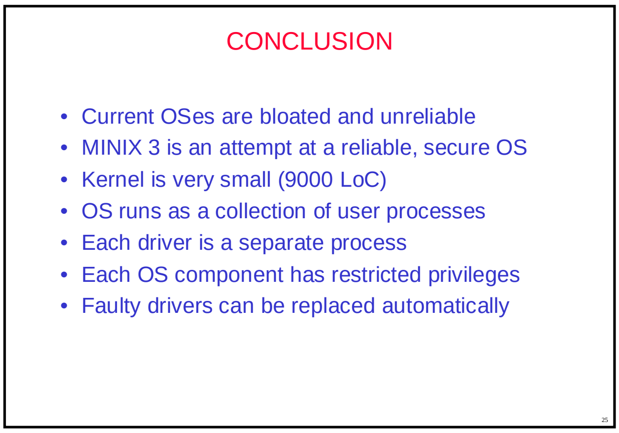
- We injected 800,000 faults into each of 3 drivers
- Done on the binary drivers
- Examples, change src addr, dest addr, loop condition
- 100 faults were injected on each experiment
- Waited 1 sec to see if the driver crashed
- If no crash, inject another 100 faults and repeat
- The driver crashed in 18,038 trials
- The operating system *NEVER* crashed

LIVE UPDATE

- We are adding live update to the system
- Goal: replace any component without a reboot
- Update manager coordinates the update
- Tells the component to be updated to finish & exit
- When it is gone a new one is started
- Internal connections use a virtual endpoint

POSITIONING OF MINIX

- Show that multiserver systems are reliable
- Demonstrate that drivers belong in user mode
- High-reliability and fault-tolerant applications
- $50 single-chip, small-RAM laptops for 3rd world
- Embedded systems:
  - DVD players
  - cell phones
  - digital cameras
  - TVs
  - etc.



CONCLUSION

- Current OSes are bloated and unreliable
- MINIX 3 is an attempt at a reliable, secure OS
- Kernel is very small (9000 LoC)
- OS runs as a collection of user processes
- Each driver is a separate process
- Each OS component has restricted privileges
- Faulty drivers can be replaced automatically

25