Microkernels

Microkernels i

- Who here uses an apple device (iOS, watchOS, TvOS, Mac OS X)
- Or an Intel x86 computer?
- If you said yes to either, you're using a microkernel.
- XNU/Darwin is a hybrid kernel + BSD-like userspace for Apple products.
- Minix is used on Intel chips.

What's a Monolithic Kernel? i

- A monolithic kernel (Linux, the BSDs, old Windows, classic Mac OS) is one program (pid 0) that manages the underlying hardware for you.
- Interfacing is done with system calls; you issue a system call and then the OS will do what's required for you.
- Thus, any **user space** program (e.g. `ls`, `cat`, `firefox`) on Linux is coupled to Linux via its system call interface.
- Linux can thus never "remove" a system call, since userspace needs backwards compatibility. Thus we have `dup`, `dup2`, `dup3`, `preadv`, `preadv2`, `fstat`, `fstat64`, etc.

What's a Microkernel? i

- A Microkernel manages as little code as possible. For Minix, the kernel is ~6000 loc.
- As much as possible is put into user space, and thus, can be treated as processes.
- In a Monolithic Kernel, drivers can be externally loaded (with root), but are linked to the kernel's address space – any malicious or improperly written drivers can easily crash your computer. For a Microkernel, devices are run in user space, and thus crash like a normal process does.
- Message passing is used between user space components. Messages are sent and then actions are taken on them, rather than interfacing with the system call library.

Microkernels i