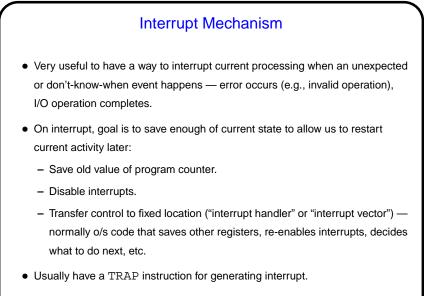
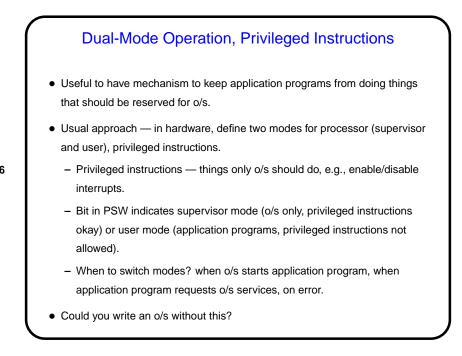
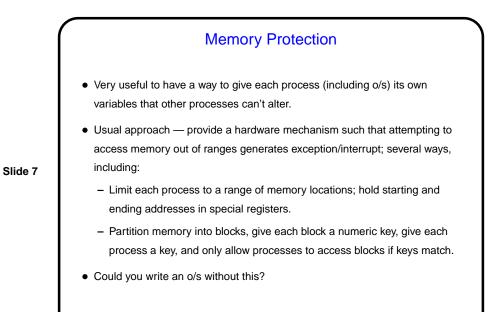


Slide 5

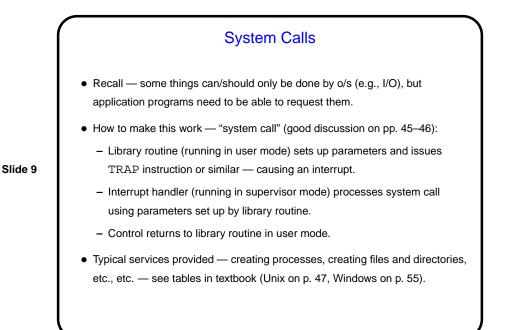


• Could you write an o/s without this?





Timer • Useful to have a way to set a timer / "alarm clock" - e.g., to get control back if application program enters infinite loop. • Usual approach — hardware features that tracks real time and can be set to interrupt CPU.



Shell
History — early batch systems had to interpret "control cards"; modern equivalent is to interpret "commands" (usually interactive).
Not technically part of o/s, but important and related.
Typical shell functionality:

Invocation of programs (optionally in background).
Input/output redirection.
Program-to-program connections (pipes).
"Wildcard" capability.
Scripting capability.

Examples — MS-DOS command . com; Unix sh, bash, csh, tcsh, ksh, zsh, ...

Minute Essay

- I once had a learning experience about "how DOS is different from a real o/s". Summary version: A program using pointers (possibly uninitialized) caused the whole machine to lock up, so thoroughly that the only recovery was to power-cycle.
 - What do you think went wrong?

Slide 11

Slide 12 • The program changed memory at the addresses pointed to by the uninitialized pointers — and this memory was being used by the o/s, possibly to store something related to interrupt handling. A "real" o/s wouldn't allow this! (Then again, the version of MS-DOS in question was supposedly written to run on hardware that didn't provide memory protection, so maybe it's not DOS's fault.)