Feb 16, 2005 -- Lecture 13

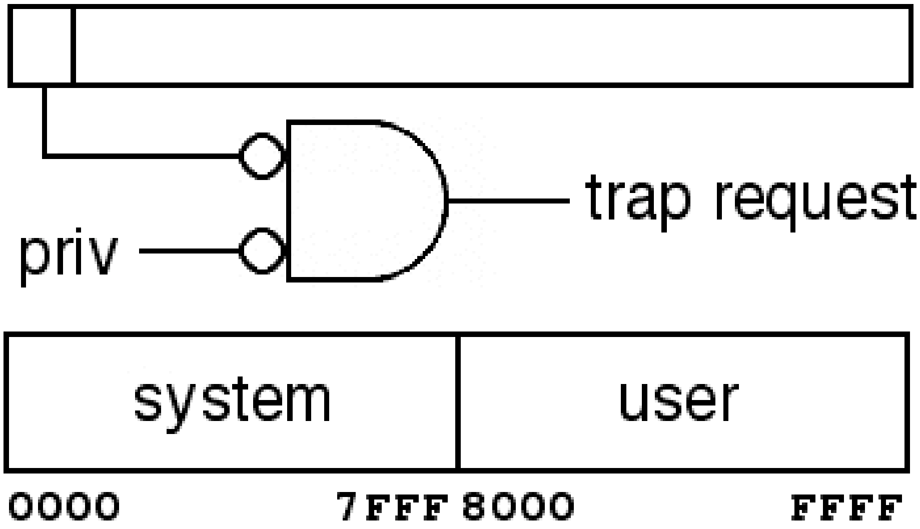
22C:169
Computer Security

Douglas W. Jones

Department of Computer Science

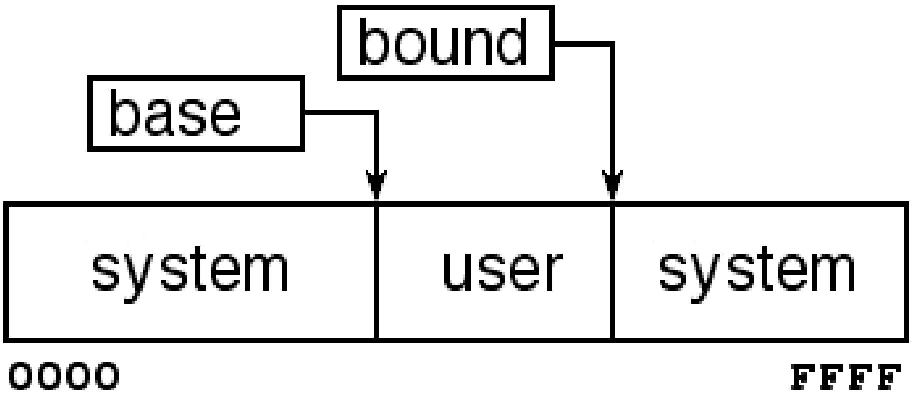
Memory Protection

Crudest memory protection idea:



This is inflexible, but it is sufficient

Generalization

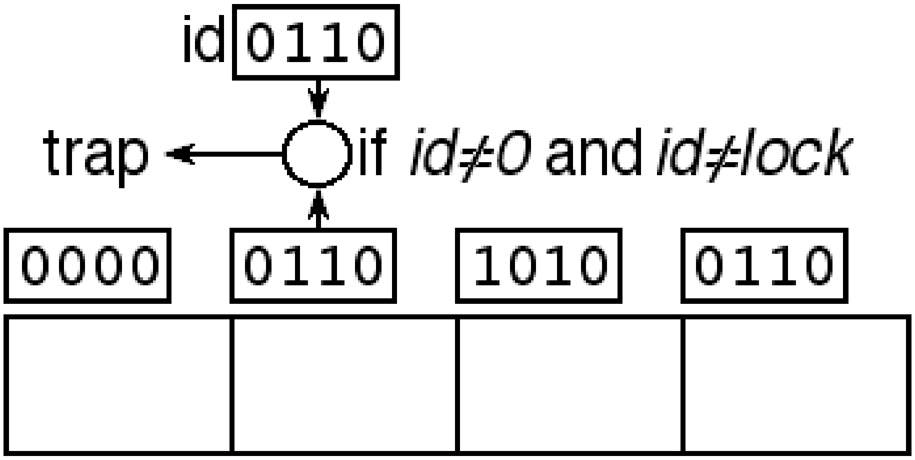


```
if ((addr < base)||(addr > bound))
    if (!privileged) trap;
```

Allows multiple users!
Requires parameter validation!

Another Generalization

IBM, 1965, System 360

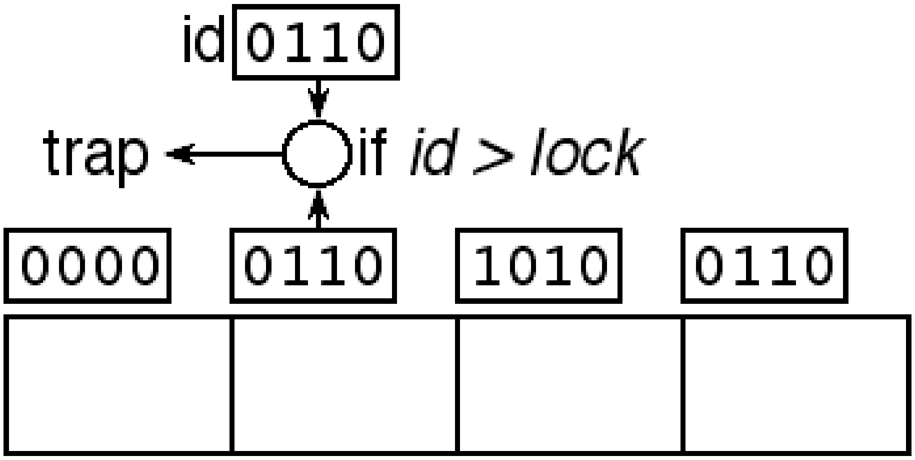


"pages" were 4k each

allows system + 15 active users

Another Generalization

MIT/GE/Bell Labs Multics, ~1965



A hierarchy of security "rings"
allows drivers+kernel+filesys+...+user

internal subdivision of system!

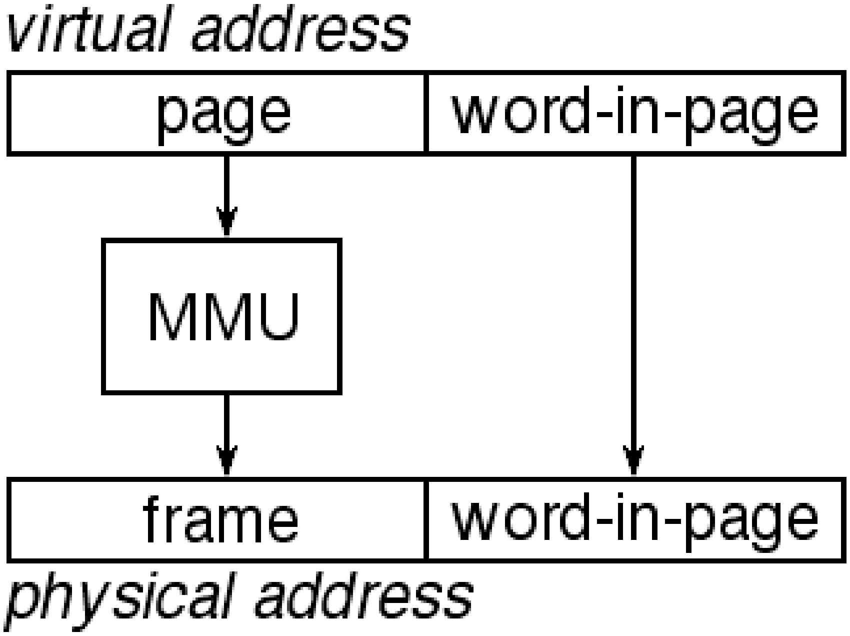
But strict hierarchies quickly fail

Paged Virtual Memory

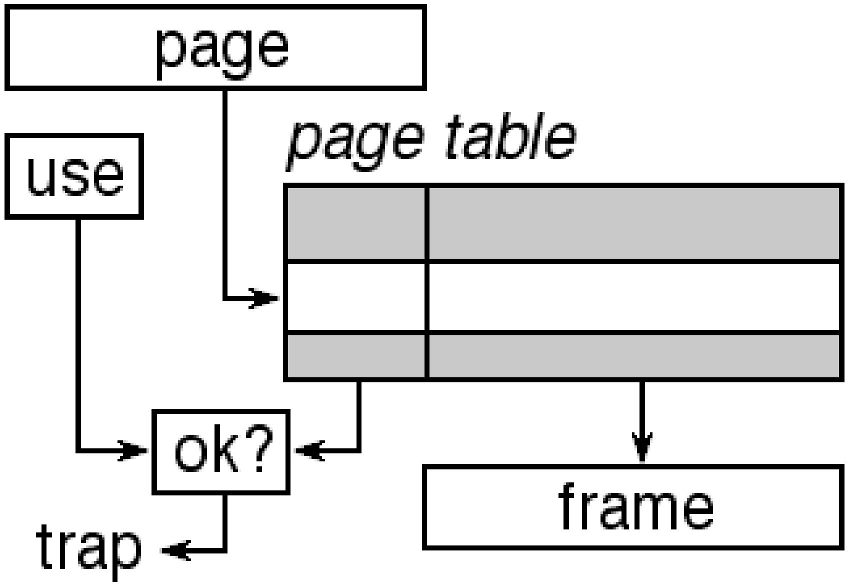
Ferranti Atlas

Manchester, 1962; proposal traced back to 1957.

CACM October 1961, John Fotheringham



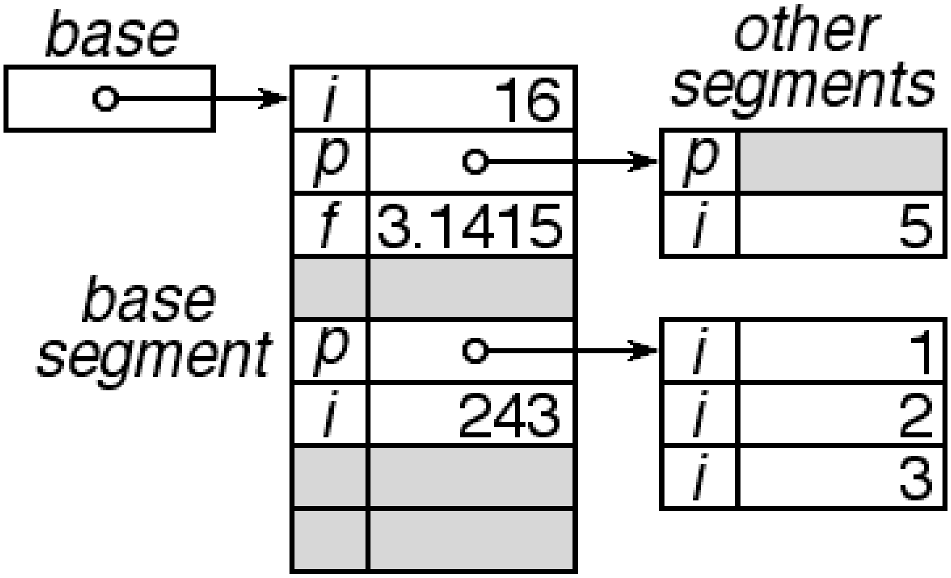
The Memory Management Unit:



Many possible implementations of idea

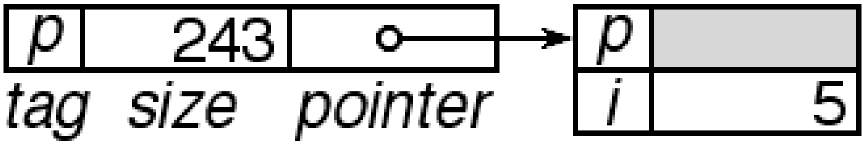
Tagged Architectures

Burroughs Corp, 1961

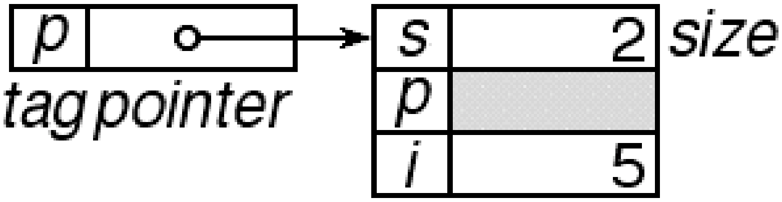


Tagged data alternatives

Original Burroughs Model

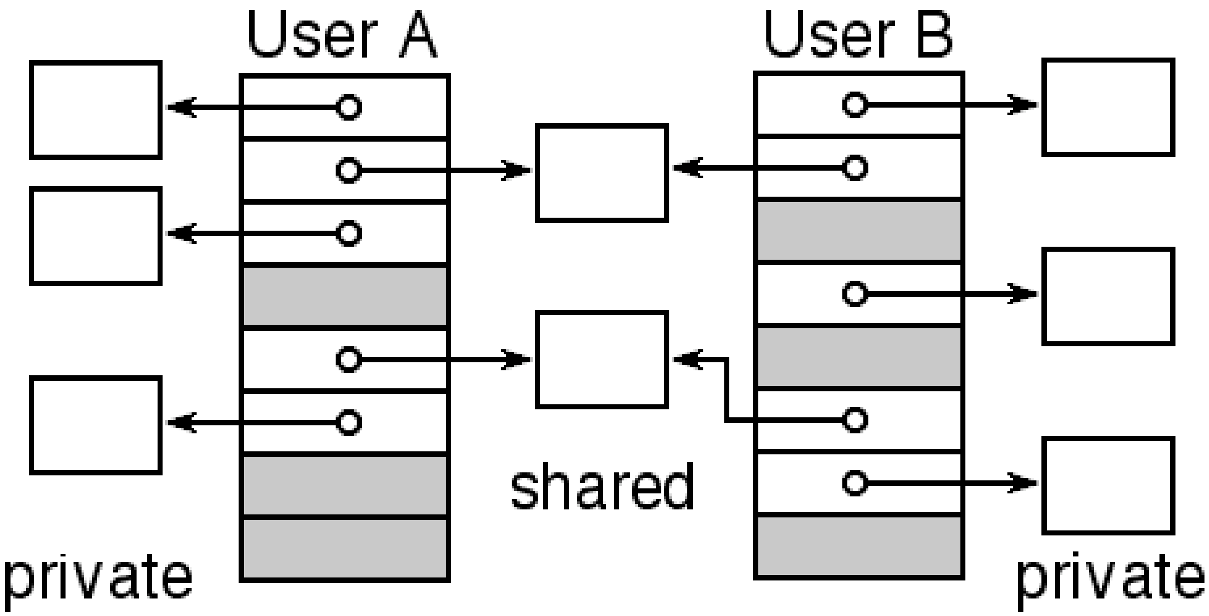


Alternative

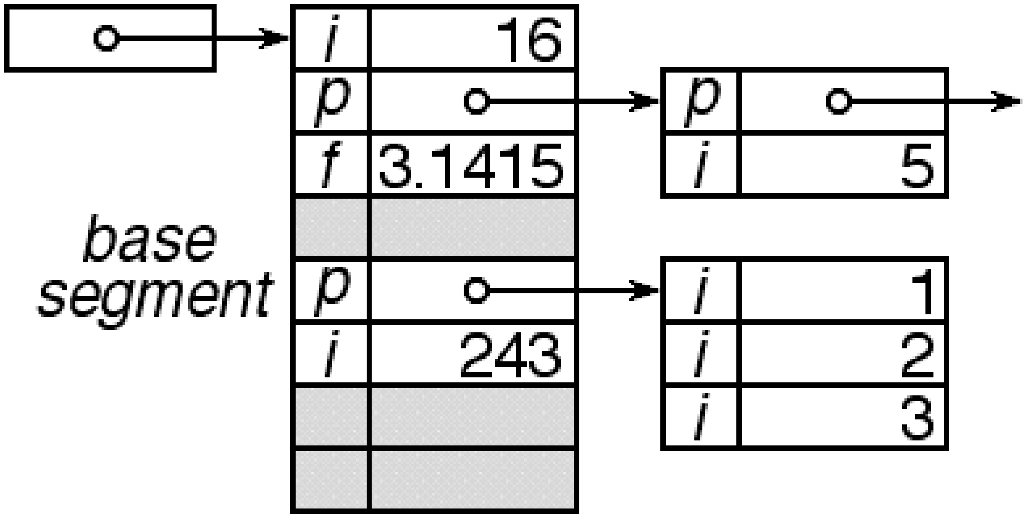


Page Table = Memory Domain!

Lampson et al, CACM Dec. 1966

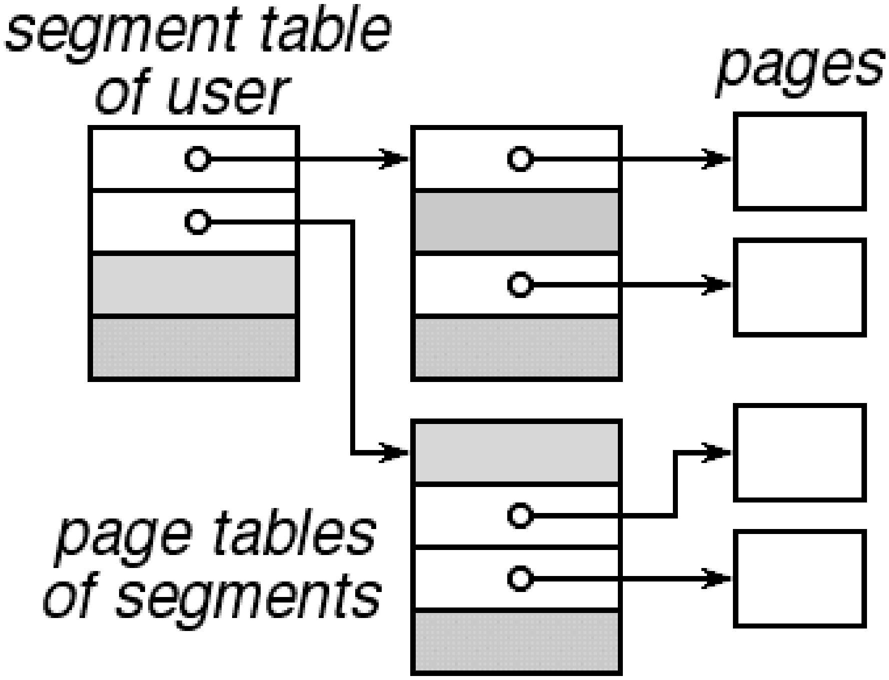


Transitive Closure = Memory Domain



The Multics Memory Model

born 1965, in service, 1968 final system shutdown in 2000



The Multics Memory Model II

Segments may be shared

- *Individual are never shared*

Segment attributes include level
Page attributes include gateway bit

Multics gate crossing

- *Legal to call to higher privilege level if*
    - Call is to word 0 of page
    - Gateway bit is set for that page
- *Result: push old level, set new level*

What's Wrong with Multics

Successfully protects

- *High privilege code from low*
- *Proprietary package from package user*
- *Debugged code from module under test*

Does not solve mutual suspicion problem

- *undebugged proprietary code*
  - must not damage or inspect caller
- *caller must be able to call but*
  - must not inspect proprietary code