Kernel Objects

Peter J. Denning

- Step back for big picture view of kernel objects

- Each kernel level is the manager of a class of objects. A few levels manage multiple related objects -- notably
  - threads and semaphores (Level 2)
  - files and buffers (Level 6)

- All present a user interface for their objects
  - Always a CREATE and DELETE operation
  - Other operations suitable for their objects

- Every object has a **pointer** to it

- The pointer maps to a **desciptor** (control block) that shows where in memory the object is, and may contain some other control information

- Microkernel pointers are simply handles generated by the levels when they create objects. No more protection is needed since microkernel software is trusted.

- User kernel pointers are a different story. They must be protected from alteration. This is done with unforgeable capabilities that are potentially sharable across the Internet.

Level 2 – Concurrency Control

- OBJECTS
  - threads, semaphores
- INTERFACE
  - CR_THR, DEL_THR, SAVESW, LOADSW, DISABLE, ENABLE, SUSPEND, RESUME, SLEEP, WAKEUP, WAIT, SIGNAL
- OPERATION OF CREATE
  - i = CR_THR gets a TCB from the TCB free list, initializes to empty stateword with IP=0 and empty successor link, returns TCB index i
  - j = CR_SEM(init) generates a SCB[j] with initial counter and empty queue
- NOTES
  - Operations SAVESW, LOADSW, DISABLE, and ENABLE are visible at interface only in kernel mode

Level 3 – Virtual Memory

- OBJECTS
  - address spaces, master files
- INTERFACE
  - CR_AS, DEL_AS, CR_MF, DEL_MF
- OPERATION OF CREATE
  - ascb = CR_AS(mf)
  - Creates an address space control block (ascb) that contains two pointers: (1) to the PT and (2) to the master file (mf) on the disk
  - pointer ascb placed in the virtual memory AS slot
  - Initial PT shows all pages missing
- NOTES
  - CR_MF is the linker-loader; it generates the address-space image on the disk and returns a pointer mf to it, which is then used in call to CR_AS.
  - master file (mf) also called ”swap file”

Level 4 – Message Passing

- OBJECTS
  - message queues, message buffers
- INTERFACE
  - CR_MQ, DEL_MQ, GET_BUFF, RETURN_BUF, SEND, REPLY, GET_REPLY
- OPERATION
  - mq = CR_MQ( ) returns pointer mq placed in the inqueue slot of virtual machine; mq points to control block with head, tail, and semaphore
  - mb = GET_BUFF returns pointer to a message buffer (wait if unavailable)
  - SEND(s,p,m) where p is sender and m a message, places ”p,m” in empty mb and attaches to inqueue[s]
  - mb = GET_IN returns the first message buffer in inqueue[self]; mb contains name of sender and message
  - REPLY(mb,R) places reply R in buffer mq.message field
  - R = GET_REPLY(mb) returns content of mb.message, releases buffer

Level 5 – Global Address Space

- OBJECTS
  - Capabilities, C-lists (CL)
- INTERFACE
  - CR_CAP, DEL_CAP, CR_CL, DEL_CL, COPY_CAP, MAKE_PUBLIC, INSERT, REMOVE
- OPERATION OF CREATE
  - i = CR_CP(t,h) entry (t,a,h) in CL[i], where a is all access
  - c = CR-CL(h) creates CL capability c = (CL,a,h) for a new empty CL; stored in VM.CL slot
- NOTE
  - http protocol for URLs not included at this level because http and URLs are serviced at Level 10 (service processes for users)
  - The created cap and CL are in kernel space and therefore protected from alteration

Level 6 – Information Objects

- OBJECTS
  - files, pipes, devices
- INTERFACE
  - CR_IO, DEL_IO, OPEN, CLOSE, READ, WRITE
- OPERATION OF CREATE
  - i = CR_IO(X) creates a new information object of type X and a UFAT handle, adds (handle,descriptor) to the hash table for X, and places the capability (X,allaccess,handle) in CL[i]
- NOTE
  - Generic interface sketched above

Level 7 – Directories

- OBJECTS
  - directories
- INTERFACE
  - CR_DIR, DEL_DIR, SEARCH, SEARCH_PATH, ATTACH, REMOVE, RENAME
- OPERATION OF CREATE
  - c = CR_DIR(p) creates an empty directory and a capability c = (D,allaccess,handle) pointing to it. Places self-capability c in the first "." entry and parent-capability p in the second ".." entry.
- NOTE
  - Capabilities in directories are in public form

Level 8 – Process Virtual Machines

- OBJECTS
  - virtual machines
- INTERFACE
  - CR_VM, DEL_VM, SUSPEND, RESUME, EXIT
- OPERATION OF CREATE
  - c = CR_VM(list-of-values) creates a virtual machine template and returns a capability c = (VM,allaccess,handle) for it. The VM slots (IN, OUT, USER, THREAD, ARGS, AS, CL, INQUEUE, PAR, SIB, CD, PATH) are filled from the list-of-values
- NOTE
  - SUSPEND, RESUME are used by parent
  - EXIT is used by VM to signal its completion

Level 9 – Shell

- OBJECTS
  - command lines
- INTERFACE
  - no externally called shell commands
- OPERATION OF CREATE
  - shell already exists as library program from system boot
- NOTE
  - shell is started by login program when user presents proper credentials

Summary

- This summary gives overview of the kernel levels showing how they create their respective objects and create the pointers to them.
- Capabilities are used at the user levels only.