Plan 9 / PPC Virtual Memory

Porting Plan 9 to the PowerPC 74xx Architecture

Ajay Surie
Adam Wolbach
15-412 Operating Systems Practicum

Kernel Address Space

- BATs provide virtualization for kernel
  - Kernel space is contiguously, but not directly mapped
    - EA base: 0x80000000, #define'd as KZERO
    - RA/physical base: 0x00000000
      - Kernel TEXT starts at base+0x1000 for both EA and RA
      - This appears to be a Plan 9 standard convention for all code
- Segment registers untouched by kernel
- Total EA range: 2GB, 8 BAT registers * 256 MB
  - Existing PPC implementation only uses 512 MB for kernel mappings, overlaying 2 pairs of I/DBATS
  - Also uses a DBAT for device communication: FPGA

User Process Address Space

- All virtualization is done through segmentation, which can map the entire effective address space, if necessary
  - Opposite of kernel's policy, BATs off-limits
  - TODO: Where do user processes EA start?
- Old PPC implementation uses 8 segment registers out of 16 available
  - Registers 8 – 15 unused, EA's given to kernel

Plan 9 /port “Segmentation”

- Each user process keeps track of a few types of “segments”, a misnomer for regions
  - TEXT, DATA, BSS, SHARED, STACK, PHYSICAL, RONLY, CEXEC, SEG1->4
- Each region obviously handled differently upon page fault

Page Faults in Regions

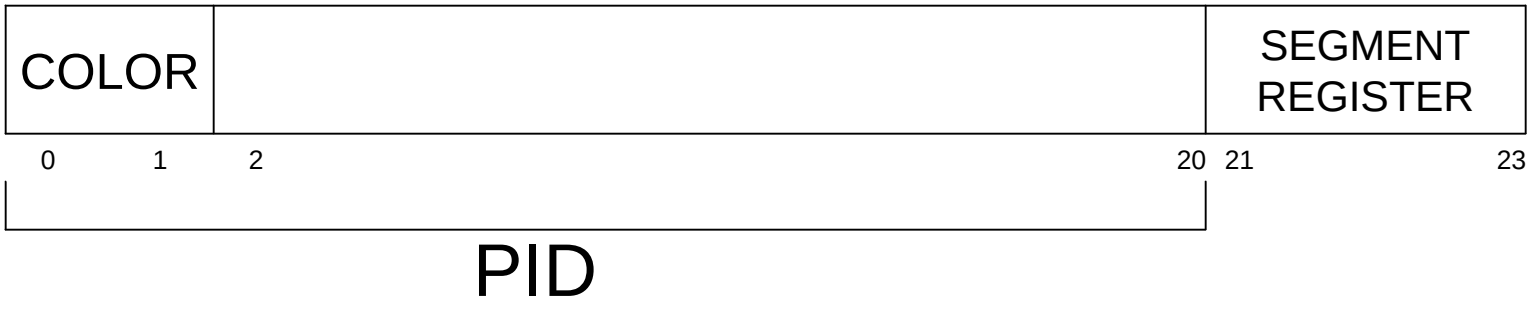
- Copy-on-write concept used extensively
  - Shared r/o: TEXT, SHARED, PHYSICAL
    - Page fault confuses the handler; kills proc
  - Zero-Filled-On-Demand: BSS, STACK
  - Generic COW: DATA
  - Not sure yet: Special Types

Page Insertion / Eviction Policy

- Current Plan 9 /ppc code doesn't use rehashing for segmented address inserts, so some hardware lookups are wasted
- If a PTEG is full (i.e. hash collision), a PTE is chosen for replacement essentially at random by a looping pointer
  - Next PTE to be replaced indicated by a global offset indexing into a PTEG
    - offset = (offset + 8) % 64; //done on eviction

VSID Construction

- Each process assigned a PID from a pool of available 21 bit PIDs
- High order 2 bits of PID make up its color
  - Used by PID reclamation algorithm
- VSID = (PID || Segment Register) : 24 bits



PID Reclamation/Revocation Algorithm

- Sweeper thread reclaims PIDs periodically
- Woken up when color of the next PID to be assigned is equal to a “trigger” color C_T
  - “Sweep” color C_S one ahead of C_T
  - For each process P, if COLOR(P) == C_T set P’s PID = 0; clear P’s page mappings
- If no PIDs left, set PID = 0 for every process, reset PID counter, flush TLB

Future Short-Term Milestones

- How segment types correlate to segment registers in P9
- TLB: EA -> RA or VA -> RA
  - Determines flush policy
- A coherent picture of main memory
- Compare /sys/src/9/mtx with /sys/src/9/ppc
- More detailed code breakdown

Sources

- The PowerPC Architecture: A Specification For a New Family of RISC Processors, Morgan Kaufmann Publishers, San Francisco, 1994
- /sys/src/9/port; /sys/src/9/ppc
  - Bell Labs; Charles Forsyth