Operating Systems Foundations (solution notes)

p11q6 JMB

(a) No segmentation hardware → process page tables are likely to map the whole of memory. Logical segments, placed in V.A. space by e.g. compilers, will appear as clusters of pages in a sparse table.

Typical page table entry will contain :-

- page number – or may be implicit and used as index into PPT.
- ~~present~~ valid – does the page exist?
- in-memory/swapped?
- page-frame (base) in memory.
- global, kernel, shared, copy-on-write, read-only, read/write, execute accessed?, clean/dirty? to support sharing, access control, memory management.

(b) (i) PTBR contains base address of current process's P.T.
PTLR contains size/length

VA: page#, offset
check page# against PTLR – is page in range?
use PTBR(page#) to reach entry for page
check valid, swapped, R/W/E ... address is base + offset.

Note 2 memory accesses per address resolution.

(ii) A TLB is a relatively small associative memory.
It has entries for the (e.g. 64) most recently accessed pages.
Typical entry: process#, page#, base/frame#, R/W/E, accessed?

VA: page#, offset
perform hardware/associative search of
process #, page # with those fields of TLB
if found check access – if OK address = base + offset
set accessed/written bits
if not found – interrupt – page fault.
trap to OS to search in-memory P.T. – find page +
make entry in TLB.
Restart instruction.

Note – no extra memory reference.

(20)

On part 2 of the course – memory management