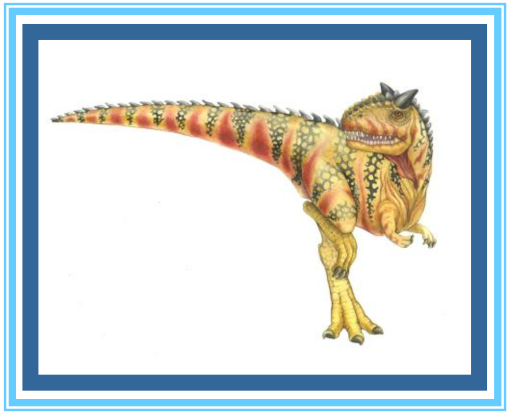
Project 2: Virtual Memory



Virtual Memory Simulation

- Aim:
  - Implement a virtual memory simulator in any high-level language (C/C++, Java, Python, …)
- Requirements:
  - 32-bit logical address
  - 24-bit physical address
  - Page (Frame) Size: 4KB
  - Page replacement algorithm: LRU
  - Assuming that the page table is stored in a separated space (not in the 24-bit physical address space)
  - Only the functional behavior is simulated. No latency simulation is required.

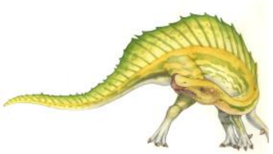
Virtual Memory Simulation

- Input trace files is given in traces.zip.
  - Five benchmarks included

- Trace file format:
  - s 0x1fffff50
    - s or l means store or load
    - 0x1fffff50 is the 32-bit logical address in hexadecimal

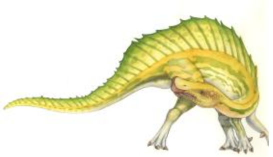
3. Virtual Memory (Option 3)

- Results Requirements (Outputs for all five benchmarks):
  - Number of page faults
  - Page Table hit rate
    - Hit rate = number of page hit / number of page reference



Submission Requirements

- Deadline: 11:59 pm, May 4th 2010 (Monday, last day of class)
- Teamwork allowed (no more 2 students in one group)
- What to submit:
  - A project report on your design and implementation.
  - Include the screen shots of the test results and outputs in your project report.
  - Include all your implementation source codes.



Extra Credits (5 points)

- Implement FIFO page replacement algorithm

- Compare page table hit rates of the LRU and FIFO algorithms for the five benchmarks.

- Which algorithm is better according to your simulation results? Give your own thoughts (explanations) on the results.