

COMP2300

Tutorial / Laboratory 08 - Virtual Memory in PeANUt

Semester 1, 2007

Week 9 (30 Apr - 4 May)

Draft (some bugs in Peanut VM have been exposed!)

Note that for this session, there are some [Preparation Exercises](#). Also for this session, there is a submittable laboratory exercise which is due by 10 am Monday 14 May (week 10), which will contribute up to 1% of your assessment (in the Tute/Lab mark).

Objectives

There are several objectives in this exercise:

- To deepen your understanding of procedures, traps and bit operations in PeANUt.
- To improve your understanding of how virtual memory (paging) is implemented on the PeANUt machine.
- To improve your understanding of how the LRU and FIFO paging policies can be implemented, and of the paging characteristics of each.

Preparation

NOTE: This laboratory requires you to read the detailed description of the operation of **PeANUt Virtual Memory** that appears in **Section 3** of the **PeANUt Specification**. It is highly desirable that you read this in advance of your laboratory. Reading the rest of this document in advance will be helpful too. You will waste your lab time and miss out on opportunities to ask your tutor relevant questions if you do not prepare adequately.

Preparation Exercises

Complete the following questions on a separate sheet of paper, with your name and student number clearly written. Please ensure your writing is legible. Hand in to your tutor at the *beginning of your tutorial / laboratory session*.

1. What is meant by the terms *page* and *page frame*?
2. What is a *page fault*? When does it happen?
3. What is the role of the *dirty bit*? Where is it found?
4. What is *thrashing*?

Tutorial Exercises

1. What are the possible chains of events that can happen during a `load` instruction between an address being loaded into the **MAR** and the data appearing in the **MDR**?
2. What sequence of paging events take place when a program is first being run?
3. It is said that the LRU page replacement policy is far better than the FIFO. What does this statement mean? Why would we expect it to be so?
4. Suppose page 8_{10} is present in physical memory page frame 5 and a `store` to address `a401`. Suppose also that 3 page faults have occurred since it was first loaded into physical memory. Write the corresponding page table entry for the page, in terms of a diagram and in terms of a binary and hexadecimal number.
5. What is the largest working set (in number of pages as well as number of cells) possible for a Peanut program before the system starts thrashing? *Hint: how many pages must be reserved for VM?*