Paging Summary - SistOp

Lautaro Bachmann

Contents

Paging: Introduction

paging,

Paging refers to chop up space into fixed-sized pieces. each of which we call a **page.**

physical memory

an array of fixed-sized slots called **page frames;** each of these frames can contain a single virtual-memory page.

Overview

Paging advantages

flexibility:
Probably the most important improvement is flexibility.

With a fully-developed paging approach, the system will be able to support the abstraction of an address space effectively, regardless of how a process uses the address space;

simplicity
When the OS wants to place the adress space into physical memory it simply has to find some free pages to use;

perhaps the OS keeps a **free list** of all free pages for this, and just grabs the first free pages off of this list.

page table.

To record where each virtual page of the address space is placed in physical memory, the operating system usually keeps a per-process data structure known as a page table.

major role
store **address translations** for each of the virtual pages of the address space, thus letting us know where in physical memory each page resides.

important to remember
this page table is a **per-process** data structure

Translate a virtual adress

To translate a virtual address that the process generated, we have to first split it into two components: **the virtual page number (VPN),** and the **offset**

Where Are Page Tables Stored?

Page tables can get terribly large,

we store the page table for each process in **memory** somewhere. Let's assume for now that the page tables live in physical memory that the OS manages;

What's Actually In The Page Table?

page table organization.

The page table is just a data structure that is used to map virtual addresses to physical addresses. Thus, any data structure could work.

linear page table,

The simplest form is called a linear page table, which is just an array.

The OS **indexes** the array by the **virtual page number** (VPN), and looks up the **page-table entry** (PTE) at that index in order to find the desired **physical frame number** (PFN).

contents of each PTE,

A valid bit
is common to indicate whether the particular translation is valid; All the unused space in-between the address space will be marked **invalid,**

by simply marking all the unused pages in the address space invalid, we remove the need to allocate physical frames for those pages and thus save a great deal of memory.

protection bits,
indicating whether the page could be read from, written to, or executed from.

present bit
indicates whether this page is in physical memory or on disk

dirty bit
indicating whether the page has been modified since it was brought into memory.

reference bit (a.k.a. accessed bit)
is sometimes used to track whether a page has been accessed, and is useful in determining which pages are popular and thus should be kept in memory;

Paging: Also Too Slow

With page tables in memory, we already know that they might be too big. As it turns out, they can slow things down too.