

Laptop topics

Security features

- login / password.
- firewalls
- User account Control
- buffer overrun attacks

Page replacement Algorithm.

Last Frequently used

- needs to timestamp every page accessed.

hardware Solutions

- equip hardware with 64 bit counter for each page / frame
- Increment counter each time page referenced.

Another hardware solution

- Maintain a matrix $n \times n$
- initialised to $n - number = 0$
- when page K is referenced.
- set all bits row K to 1
- set all bits of col K to 0

- Row with lowest number is least frequently used.

ref string 0, 1, 2, 3, 2, 1, 0, 3, 2, 3

4 frames

0 0 0 0	0 1 1 1	0 0 1 1	0 0 0 1
0 0 0 0	0 0 0 0	1 0 1 1	1 0 0 1
0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Not frequently used

Approximation to least frequently used

- Software counter for each page we initialize to zero at each timer interrupt O.S. scans all pages and add the reference bit to counter. When page fault occurs pick page with lowest counter value is picked
- Does not always get the exact count that the page was accessed.
- that one guy won't stop talking.

NFU with ageing

- a timer interrupt
- Counter shifted to right 1 bit
- add the R bit on the left.

0 0 0 0 0

1 0 0 0 0

1 1 0 0 0

0 1 1 0 0

0 0 1 1 0

1 0 0 1 1

List of what we have done

Optimal

fifo

Clock

LRU

NFU

NFU with ageing.

NRU

Not recently used - NRU

when a page fault occurs

O.S. inspects all the pages

divide into 4 categories

0 - not referenced, not modified

1 - not referenced, modified.

2 - referenced , not modified

3 - referenced , modified

choose page from lowest numbered category first.

Scope

- is scope global or local

page buffering - Vax / VMS O.S.

~ variable allocation, local replacement policy

page replacement is fifo

• to improve performance replaced page is not lost

• put into two lists

- free page list if it is not modified

- modified page list.

page fault occurs if requested page is still present and one of these two lists.

remove from list. give to process effectively increasing memory allocation.

Demand cleaning - writing modified pages to disk
pre cleaning - write things to disk that are modified.

Demand Paging -

11/21/18

Day 27

Working set strategy

working set - pager program is currently accessing

- PFF - page fault frequency algorithm
- use bit associated with each page
 - Set to 1 when page is accessed.
 - when page fault occurs operating system looks at time of last page fault.
 - there should be a value f defined
 - if elapsed time $< f$
a page is added to resident set
 - if elapsed time greater than f discard all pages with use bit = 0
 - reference bit cleared to 0 at timer interrupt

* We will not get rid of the old working set until the

new working set is finished

VSWS - Variable Interval Sampled Working Set Policy. - fix having two working sets

- Use Sampling period based on virtual time
- at beginning of sampling period all are bits are reset. on resident pages.
- at the end all pages of one bit not set are discarded/expelled.

3 Parameters

M = minimum duration of Sampling interval

L = maximum duration / range

Q = number of page faults allowed to occur between sampling intervals.

VSWS policy

1. if virtual time since last sampling interval $\leq L$, suspend process, scan use bits
2. prior to L we have Q page faults occur.
 - a. if time $< M$ wait until M , suspended processor, scan use bits.
 - b. if time $\geq M$, suspend & scan otherwise add page to resident set.

Test Question

When you first load a process how much memory is allocated,

Memory allocation

- equal allocation.

of frame

of processes.

- Proportional allocation

- give what is asked for

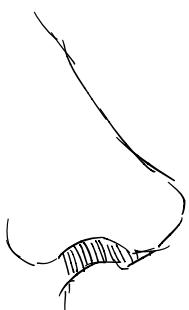
Demand paging -

I/O Locked process

page file - ledger of pages not memory resident

Load Management - how many processes do you want?

degree of multiprogramming.



Denning's criterion - " $L = S$ criterion"

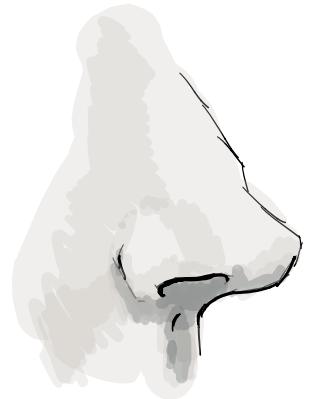
Set multiprogramming level so the mean time

between page faults = time required to

service page fault.

- Next time Q&A stuff + Hw! ;)

;))



So Nose-y!