

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Basic Concepts

Assignment Project Exam Help

- Memory Allocation
- Swapping

Virtu

Pagi

<https://eduassistpro.github.io>

- Demand Paging
- Page replacement algorithms
- Working set model

Add WeChat edu_assist_pro

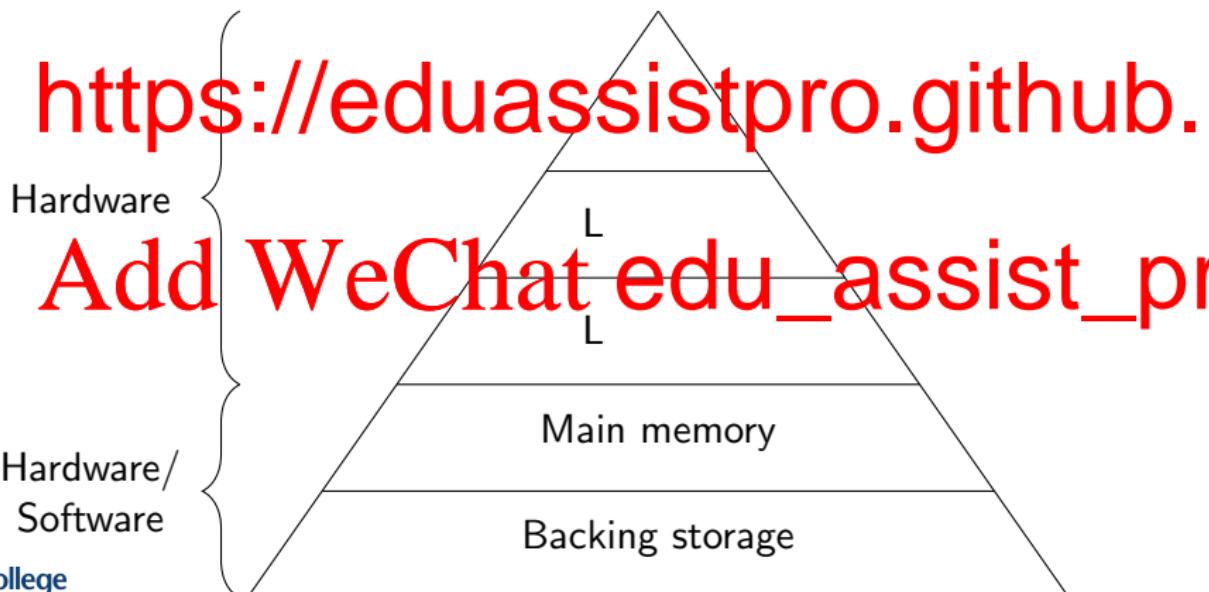
Linux Memory Management

Memory Hierarchy

Hardware: CPU registers and main memory

- Register access in one CPU clock cycle (or less)
- Main memory can take many cycles
- Caches sit between main memory and CPU registers

Managed by

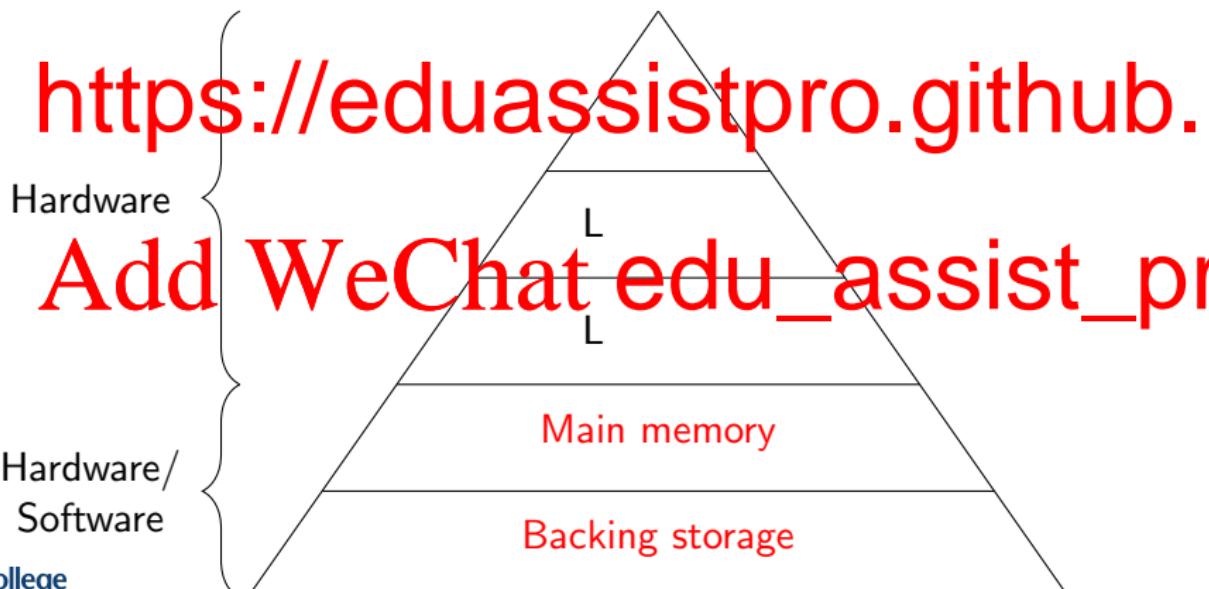


Memory Hierarchy

Hardware: CPU registers and main memory

- Register access in one CPU clock cycle (or less)
- Main memory can take many cycles
- Caches sit between main memory and CPU registers

Managed by



Memory Management

Memory is a key component of the computer

- e.g. every instruction cycle involves memory access ⇒ process has to be loaded into memory before it can execute

Assignment Project Exam Help

Memory management needs to provide

-
- <https://eduassistpro.github.io>

Characteristics

- No knowledge of how memory addresses are used
 - e.g. instruction counter, indexing, indexing
- No knowledge what memory addresses are
 - e.g. instructions or data
- True for simple case but may want protection with respect to read, write, execute, etc.

Logical vs. Physical Address Space

Memory management binds logical address space to physical address space

Logical address

- Generated by the CPU
- Address space seen by the process

Phy

- <https://eduassistpro.github.io>
- Refers to physical system memory

Logical and physical addresses

- Same in compile- and load-time address-binding
- Different in execution-time address-binding schemes

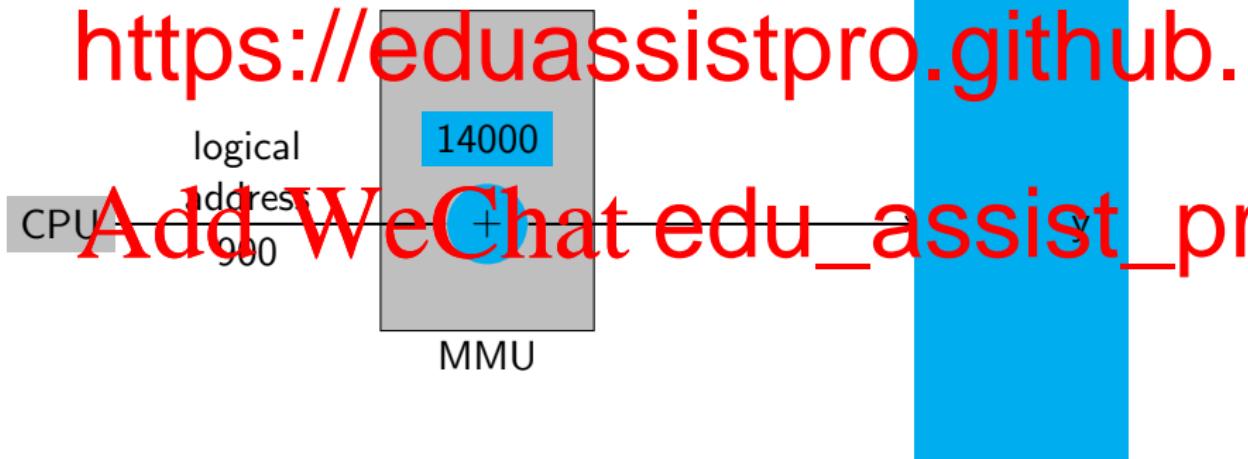
How do you achieve this mapping?

Memory-Management Unit (MMU)

Hardware device for mapping logical to physical addresses

- e.g. add value in relocation register to every address generated by process when sent to memory
- User process deals with logical addresses only
- Has to be fast implemented in hardware

<https://eduassistpro.github.io>



Main memory is usually split into two partitions:

- Resident operating system (kernel)

- Usually held in low memory with interrupt vector

-

<https://eduassistpro.github.io>

How do you decide where to load a new process?

Need to figure out the strategy for process to be loaded at correct location

Add WeChat edu_assist_pro

Assignment Project Exam Help

- **base** register contains physical start address for process

•

<https://eduassistpro.github.io>

- Physical address = logical address + **bas**

If logical address > limit, then error

Add WeChat edu_assist_pro

`base` and `limit` register define logical address space

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

e.g. jmp 100 in program would go to physical location 300140

Hole

- Block of available memory
- Holes of various size scattered throughout memory

Assignment Project Exam Help

When new process arrives.

- allocate memory from hole large enough

OS m

: <https://eduassistpro.github.io>

Add WeChat edu_assist_pro

What is the best algorithm for allocation?

First-fit → Allocate first hole that is big enough

Best-fit → Allocate smallest hole that is big enough

- Must search entire list, unless ordered by size
-

Wor

<https://eduassistpro.github.io>

-
- Produces largest leftover hole

Why best-fit or worst-fit?

First-fit and best-fit better than worst-fit in terms of speed and storage utilisation

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

External fragmentation → memory exists to satisfy request, but not contiguous

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`

Reduce external fragmentation by compaction

- Shuffle memory contents to place all free memory together in one large block → leads to I/O bottlenecks

Problem: Number of processes limited by amount of available memory

Assignment Project Exam Help

- But ... only running processes need to be in memory

Solu

• <https://eduassistpro.github.io>

- Bring back into memory for continued exec
- Requires swap space → can be file or disk
- Transfer time is major part of swap time

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

What if a process is “too large” to fit into memory ⇒ can only part of a process exist in memory?

Assignment Project Exam Help

Separation of user logical memory from physical memory

- Only part of process needs to be in memory for execution
 - <https://eduassistpro.github.io>
 - Address spaces can be shared by several processes
 - Allows for more efficient process creation
- Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Assignment Project Exam Help

Virtual memory can be implemented via

• <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Paging

Physical address space of process can be noncontiguous

- Process allocated physical memory when available

- Avoid external fragmentation

- Avoid problems of variable sized memory chunks

Assignment Project Exam Help

Fra

:<https://eduassistpro.github.io>

Pages

- Block of same size (as frame) of l

To run program of size n pages

- Find n free frames and load program
- Set up **page table** to translate logical into physical addresses

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

How does logical address translate to physical address?

Hint:

page

Addr

https://eduassistpro.github.io

- only need to translate page number into its corresponding frame address

Add WeChat edu_assist_pro

How do you calculate the page number?

Depends on address size and page/frame size

e.g. C

: <https://eduassistpro.github.io>)

For a 10-bit virtual address we have:

- Add WeChat `edu_assist_pr`
- page offset requires 6 bits (based on above)
- page number has 4 bits (remaining bits) \Rightarrow between 0 ... 15

Page number (p)

- Used as an index into page table
- Page table has base address of pages in physical memory

Pag

: <https://eduassistpro.github.io>

For given address size of m-bits and page size of 2ⁿ

Add WeChat edu_assist_pro

page number	pa
p	d
(m - n) bits	n bits

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Address Translation

Consider a 32-bit virtual memory address and a page size of 1 KB.

How many pages can a process potentially have?

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Address Translation

Consider a 32-bit virtual memory address and a page size of 1 KB.

How many pages can a process potentially have?

1 KB/p

Num

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Address Translation

Consider a 32-bit virtual memory address and a page size of 1 KB.

How many pages can a process potentially have?

1 KB p

Num

<https://eduassistpro.github.io>

For a 32-bit address you have:

- page offset requires 10 bits
- page number has 22 bits $\Rightarrow 2^{22}$ (

Add WeChat edu_assist_pro

Internal fragmentation → Allocated memory is larger than requested memory, but size difference internal to partition

Assignment Project Exam Help

Example - Calculating Internal Fragmentation

Page

Num

Byte

Internal fragmentation = $2048 - 1086 = 962$ bytes

Add WeChat edu_assist_pr

Worst-case fragmentation ⇒ 1 frame = 1 byte

Average-case fragmentation ⇒ $\frac{1}{2}$ frame size

Assignment Project Exam Help

Are s



<https://eduassistpro.github.io>

architectures support variable page sizes

Add WeChat edu_assist_pro

Page table kept in main memory

Assignment Project Exam Help

- Page-table base register (PTBR) points to page table

- Context switch requires update of PTBR for new process page

• <https://eduassistpro.github.io>

Problem

- Inefficient as every data/instruction access needs two memory accesses → one for page table and one for data/instruction

Add WeChat edu_assist_pro

Solution: use special fast-lookup hardware cache as associative memory

Associative memory → supports parallel search

Assignment Project Exam Help

- Called Translation Look-aside Buffer (TLB)

<https://eduassistpro.github.io>

9	5
7	

Add WeChat edu_assist_pro

- Address translation (p, d)
 - If p in associative register, get frame # out
 - Otherwise, get frame # from page table in memory

Assignment Project Exam Help

TLBs usually needs to be flushed after context switch

- Can lead to substantial overhead
-

Som <https://eduassistpro.github.io>

- Uniquely identifies each process to provide protection for that process

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Example Problem

Effective Access Time

TLB Lookup = ϵ (can be < 10% of memory access time m)

Hit Ratio = α

- Fraction of times that page is found in associative registers

-

Effect

= $2m$

<https://eduassistpro.github.io>

Consider $\alpha = 80\%$, $\epsilon = 10 \text{ ns}$ for TLB se

memory access

Add WeChat edu_assist_pr

- EAT = $110 \times 0.80 + 210 \times 0.20 = 130$

A more realistic hit ratio might be 99%

- EAT = $110 \times 0.99 + 210 \times 0.01 = 111 \text{ ns}$

Why do we need need to worry?

Page table can grow to be very large in size

On a 32-bit machine with a 4 KB page size:

$\frac{32}{}$



• <https://eduassistpro.github.io>

- Size of page table = $2^{32} \times 32$ bits = 4 MB

On 64-bit machine with 4 KB pages →

Add WeChat edu_assist_pro.com

- with 8 bytes per entry, that's 30 million GB

- lot of memory to be allocated 😞

Assignment Project Exam Help

Hierarchical page table

Has

<https://eduassistpro.github.io>

Inverted page table

Add WeChat edu_assist_pro

Hierarchical Page Table I

Idea: Let the page-table be broken-up and paged if it is too large

Simple technique → **two-level page table** for a machine with
32-bit addresses and a 4 KB page size

- page offset needs 12 bits
-

How d

<https://eduassistpro.github.io>

Each part of the page table that is being paged must _____

- Add WeChat edu_assist_pr
- Recall: page size = 4 KB
- Number of entries on one page = $\frac{\text{Page size}}{\text{Address size}} = \frac{4 \text{ KB}}{32 \text{ bits}} = 2^{10}$
- No of bits required for 2^{10} entries = 10
- Address bits left for top-level page table = $32 - 10 - 12 = 10$

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Fix outer page table in memory

Two-Level Paging |

Logical address divided

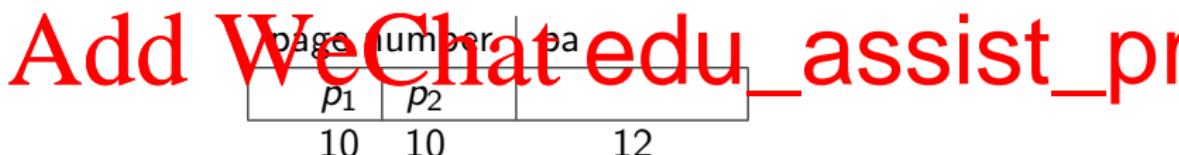
- Page number consisting of 20 bits
- Page offset consisting of 12 bits

Assignment Project Exam Help

Since page table paged, page number further divided

- <https://eduassistpro.github.io>

Thus, logical addresses as follows



$p_1 \rightarrow$ index into the outer page table

$p_2 \rightarrow$ displacement within page pointed to by outer page table

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Page Table Addressing

Consider a paging system that uses a three-level page table.

Virtual addresses are composed into four fields (v, b, γ, d) with d being the offset. What is the maximum number of pages in a virtual

Ans <https://eduassistpro.github.io>

the ad

Add WeChat edu_assist_pro

Example Problem

Page Table Addressing

Consider a paging system that uses a three-level page table.

Virtual addresses are composed into four fields (x, b, γ, d) with d being the offset. What is the maximum number of pages in a virtual

Ans
the ad

Add WeChat edu_assist_pr

Page Table: Another Idea

Don't store entry per **page** but per **frame**

- Hashed page table
- Inverted page table

Hashed Page Table

Hash virtual page number into page table

- Page table contains chain of elements hashing to same location
- Search for match of virtual page number in chain
- Extract corresponding physical frame if match found

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Inverted Page Table

One entry per physical frame

Decreases memory needed to store page table

- But increases time to search table when page reference occurs

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Paging gives one-dimensional virtual address space → what about separate address spaces for code, data, stack?

Segment

Assignment Project Exam Help

- Independent address space from 0 to some maximum

-

-

- Segment corresponds to program, procedure, array, etc.

Add WeChat edu_assist_pr

Memory allocation harder due to variable size

- May need to move segment which grows
- May suffer from external fragmentation
- But good for shared libraries

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

- One bit in table indicates whether segment is in memory
- Another bit indicates whether segment is modified

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Most OSs use only paging

Protection bits → associated with a frame indicate read-only, read-write, execute-only

Assignment Project Exam Help

Valid-invalid bit

- <https://eduassistpro.github.io>
- backing store

Add WeChat edu_assist_pro

Page replacement bits → to indicate if p or referenced (used later). Also, lock bit to prevent p being transferred out

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

When do you bring the page into memory?

Bring page into memory only **when needed**

Assignment Project Exam Help

- Lower I/O load

-

• <https://eduassistpro.github.io>

Page needed → reference it

- Invalid reference → abort
- Not-in-memory → bring into memory

Many page faults when process first starts

Eventually required pages are in memory so page fault rate drops

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Assignment Project Exam Help

Use valid-invalid bit to check memory validity

- <https://eduassistpro.github.io>
- If 0 during address translation

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

First reference, trap to OS → page fault

Assignment Project Exam Help

OS looks at another table to decide

- Invalid reference abort
-

To ha

st
<https://eduassistpro.github.io>

- Get empty frame
- Swap page into frame
- Reset tables, validation bit = 1
- Restart last instruction

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Page Fault Rate (p), $0 \leq p \leq 1.0$

If $p = 0$, no page faults
If $p = 1$, every reference causes a page fault

Eff

<https://eduassistpro.github.io>

+ [swa

+ swa

+ tests

Add WeChat `edu_assist_pro`

Note: no need to swap page out if not modified

Copy-on-Write (COW)

- Allows parent and child processes to initially share same pages in memory → if either process modifies shared page, then copy page
- Efficient process creation: copy only modified pages
-

Me

<https://eduassistpro.github.io>

- Map file into virtual address space using page tables
- Simplifies programming model for I/O

Add WeChat edu_assist_pro

I/O Interlock

- Pages must sometimes be locked into memory
 - Pages used for DMA from disk

Example Problem

Demand Paging

Memory access time = 200 ns

Average page-fault service time = 8 ms

$$\begin{aligned} \text{EAT} &= (1 - p) \times 200 + p \times (8 \text{ ms}) \\ &= (1 - p) \times 200 + p \times 8,000,000 \end{aligned}$$

If one access slow

If we want performance degradation < 1

$$\text{EAT} \leq \text{EAT} + 10\% \text{ of EA}$$

$$200 + 7,999,800 \times p \leq 220$$

$$7,999,800 \times p \leq 20$$

$$p \leq 0.0000025$$

Less than one page fault in every 400,000 memory accesses

Page Replacement

No free frame? Replace page

How do you decide which page to replace?

Find some unused page in memory to swap out → need strategy
for page replacement

Mini

<https://eduassistpro.github.io>

Prevent over-allocation of memory

Add WeChat edu_assist_pr

Use modify (dirty) bit to reduce overhead of page transfers

- Only modified pages written to disk

- ① Find location of desired page on disk

Assignment Project Exam Help

- ② Find free frame

③

<https://eduassistpro.github.io>

- ④ Load desired page into (newly) freed frame

- ⑤ Update page and frame tables

- ⑥ Restart process

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

How do

Use a
and ca

E.g. 1, 2, 3, 3, 2, 4, 1, 4, 5, 5, 7, 2, 3, 1

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Optimal Algorithm

Replace page that will not be used for the longest period of time

- Unimplementable, as knowledge of future references needed
- Useful for measuring how well other algorithms perform

Assignment Project Exam Help



<https://eduassistpro.github.io>

Reference string

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	0	1
0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1

Total of 9 page faults

Assignment Project Exam Help

- May replace heavily used page

Ref

7	0	2	0	2	0	4	2	0	1	2	1	0	1	7	1	1
7	7	7	2	2	2	4	4	4	4	0	0	0	0	7	7	7
0	0	0	3	3	3	2	2	2	1	1	1	1	1	1	0	0
1	1	1	0	0	0	0	3	3	3	2	2	2	2	2	2	1

Total of 15 page faults

Add WeChat edu_assist_pr

Heavily used pages, 0, 2, 3 are being swapped in and out

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help

Assume 3 frames



<https://eduassistpro.github.io>

Assume 4 frames

Add WeChat edu_assist_pro



Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>

Assume 4 frames
Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>

Assume 4 frames
Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

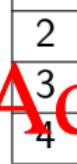
Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

Add WeChat edu_assist_pro

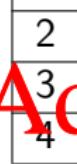
Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

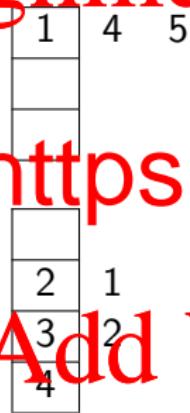
<https://eduassistpro.github.io>



Assume 4 frames

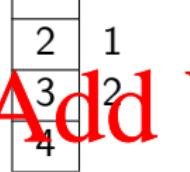
Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help

1 4 5

Assume 3 frames

<https://eduassistpro.github.io>

2 1
3 2
4 3

Assume 4 frames

Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help

1 4 5

Assume 3 frames

<https://eduassistpro.github.io>

2 1
3 2
4 3

Assume 4 frames

Add WeChat edu_assist_pro

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

10 page faults

Add WeChat edu_assist_pro

Belady's Anomaly

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (FIFO replacement)

Assignment Project Exam Help



Assume 3 frames

<https://eduassistpro.github.io>



Assume 4 frames

10 page faults

Add WeChat edu_assist_pro

Belady's Anomaly: More frames \Rightarrow more page faults

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Least Recently Used (LRU) Algorithm

Each page entry has a counter

- When page referenced, copy clock into counter
- When page needs to be replaced, choose lowest counter



<https://eduassistpro.github.io>

Reference string

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		4	4	4	0			1		1	1			
0	0	0			0		0	0	3	3			3		0		0		
1	1				3		3	2	2	2			2		2		7		

Total of 12 page faults

Proper LRU is expensive → use approximations instead

Reference bit

- With each page associate reference bit r , initially $r = 0$
- When page referenced, set $r = 1$
- ⋮

: <https://eduassistpro.github.io>

Clock Replacement Policy

- Add WeChat edu_assist_pro
- Needs reference bit r and uses clock replace
- If page to be replaced (in clock order) has $r = 1$ then
 - Set $r = 0$ and leave page in memory
 - Continue till you find $r = 0$, and replace that page
 - If all $r = 1$, replace starting page

When page fault occurs, the page being pointed to is inspected

- If $r = 0$, evict page
- If $r = 1$, clear r , and advance pointer

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Keep counter of number of references made to each page

Assignment Project Exam Help

LFU (least frequently used) algorithm

- Replace page with smallest count
-
-
- <https://eduassistpro.github.io>
-

MFU (most frequently used) algorithm

- Replace page with largest count
- Page with smallest count probably just brought in and yet to be used

Example Problem

Page Replacement

Reference string: 1, 2, 1, 3, 2, 1, 4, 3, 1, 1, 2, 4, 1, 5, 6, 2, 1.

Assuming number of frames is 3, calculate the number of page faults for LRU and Clock page replacement algorithms.

Using LRU:

1	2	1	3	2	1	4	3	1	1	2	4	1	5	6	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	2	2	2	2	2	3	3	3	3	4	4	4	4	6	6

Total of 11 page faults

Using Clock:

1	2	1	3	2	1	4	3	1	1	2	4	1	5	6	2	1
1	1	1	1	1	1	4	4	4	4	4	4	4	5	5	5	5
2	2	2	2	2	2	2	2	1	1	1	1	1	1	6	6	6
3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	1

Total of 9 page faults

For program to run efficiently

Assignment Project Exam Help

- System must maintain program's favoured subset of pages in main memory

Other

• <https://eduassistpro.github.io>

- Program repeatedly requests pages from s

Add WeChat edu_assist_pr

Locality of Reference

- Programs tend to request same pages in space and time

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Working set of pages $\rightarrow W(t, w)$

- Set of pages referenced by process during process-time interval $(t - w)$ to t

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Idea: Add “time of last use” to Clock Replacement algorithm

Assignment Project Exam Help

At each

: <https://eduassistpro.github.io>

- If age < working set age w, continue ()
- If age > working set age w
 - If page is clean, replace
 - Otherwise trigger write-back, continue to next page

Add WeChat edu_assist_pro

Working Set Size

Processes transition between working sets

- OS temporarily maintains in memory, pages outside of current working set
- Goal of memory management is to reduce mis-allocation

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

What about page fault frequency?

If many faults \Rightarrow allocate more page frames

Global vs. Local Page Replacement

Local strategy

- Each process gets fixed allocation of physical memory
- Need to pick up changes in working set size

Assignment Project Exam Help

Global strategy

- <https://eduassistpro.github.io>
- Measure page faults/per sec and increa

No universally agreed solution

- Linux: global page replacement
- Windows: local page replacement
- Depends on scheduling strategy (i.e. round-robin, ...)

Assignment Project Exam Help

Li

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Memory Management System Calls

System call	Description
s = brk(addr)	Change data segment size
a = mmap (addr, len, prot, flag, fd, offset)	Map a file/device into memory
u = munmap (addr, len)	Unmap a file/device from memory

Assignment Project Exam Help

Retu

a an <https://eduassistpro.github.io>

len

prot controls protection

flags are miscellaneous bits

fd is a file descriptor

offset is a file offset

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

On a 32-bit machine, process has 4 GB of space

Top 1 GB used for Kernel memory

- User processes can make system calls without TLB flush

•

•

<https://eduassistpro.github.io>

Kern

address space

- All memory access must be virtual but need e user memory + DMA in low memory
- Create temporary mappings for > 896 MB of physical memory in remaining 128 MB of virtual memory

Assignment Project Exam Help

- Linux memory zones
- ZONE_DMA and ZONE_DMA32: pages used for DMA
- <https://eduassistpro.github.io>
- addresses – not permanently mapped

Add WeChat edu_assist_pro

Kernel and memory map are pinned, i.e. never paged

Assignment Project Exam Help	Usually on IA-32 4 KB	On x86-64 Larger page sizes (e.g. 4 MB)
Virtual address space	levels with Physical Address Extension (PAE)	
Physical address space		
Offset bits contain page status: dirty, read-only, ...		

Offset bits contain page status: dirty, read-only, ...

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

- Tries to map contiguous pages to contiguous frames to optimise transfers
- Split and merge frames as required

Linux uses variation of clock algorithm to approximate LRU
page-replacement strategy

Assignment Project Exam Help

Memory manager uses two linked lists (and reference bits)

-

<https://eduassistpro.github.io>

- Inactive list

Add WeChat edu_assist_pro

Contains inactive pages

- Least-recently used pages near tail of ina

- Only replace pages in inactive list

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Assignment Project Exam Help

KWALD Swap Thread

- Pages in inactive list reclaimed when memory low

-

<https://eduassistpro.github.io>

pdflush kernel thread

- Periodically flushes dirty pages to disk

Add WeChat edu_assist_pro