Amdahls law:

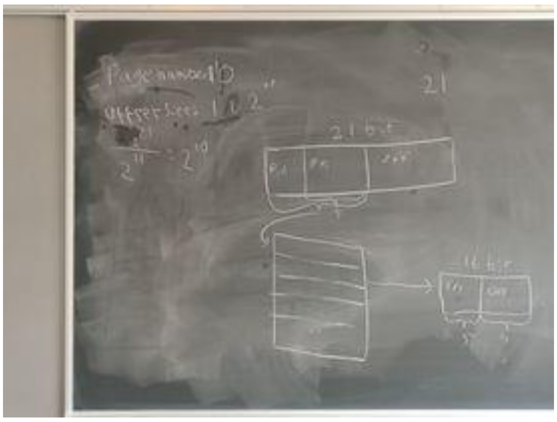
s = % of serial processing
N = number of cores

speedup = $\frac{1}{S + \frac{1-s}{N}}$ (beklager formatering lol)

Filesystems:

Begreber:

NFS: Network file system, a stateless way to mount filesystem over network
(͡° ͜ʖ ͡°)
Volume: A volume is a partition with a file system present
(͡° ͜ʖ ͡°)
Partition: A partition is a part of a harddrive or storage unit.
(͡° ͜ʖ ͡°)
Internal fragmentation: When the fragmentation happens inside the blocks, often happens with blocks of fixed sizes
(͡° ͜ʖ ͡°)
External fragmentation: When the fragmentation happens outside the blocks, often happens with variable block sizes.
(͡° ͜ʖ ͡°)
Stateless vs statefull: In a stateless file system you are unable to open and close files, as they do not save a state of where you were in a file, where a stateful makes it possible to keep files open. NFS is Stateless it also doesn’t use session-semantics
(͡° ͜ʖ ͡°)
FCB (File control block): Keeps information of a file, is the inode of assignment 3.
(͡° ͜ʖ ͡°)
Contiguous allocation: Memory blocks of variable sizes, depending on the size of the file.
(͡° ͜ʖ ͡°)
Linked/Indexed allocation: Memory blocks of fixed sizes. which is either a linked list or indexed list of all the memory blocks.



Memory:

***8.20 Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

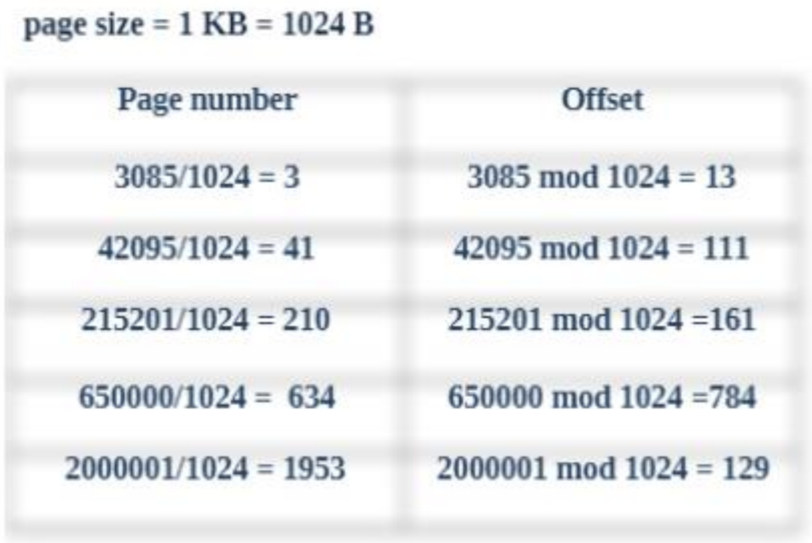
a. 3085

b. 42095

c. 215201

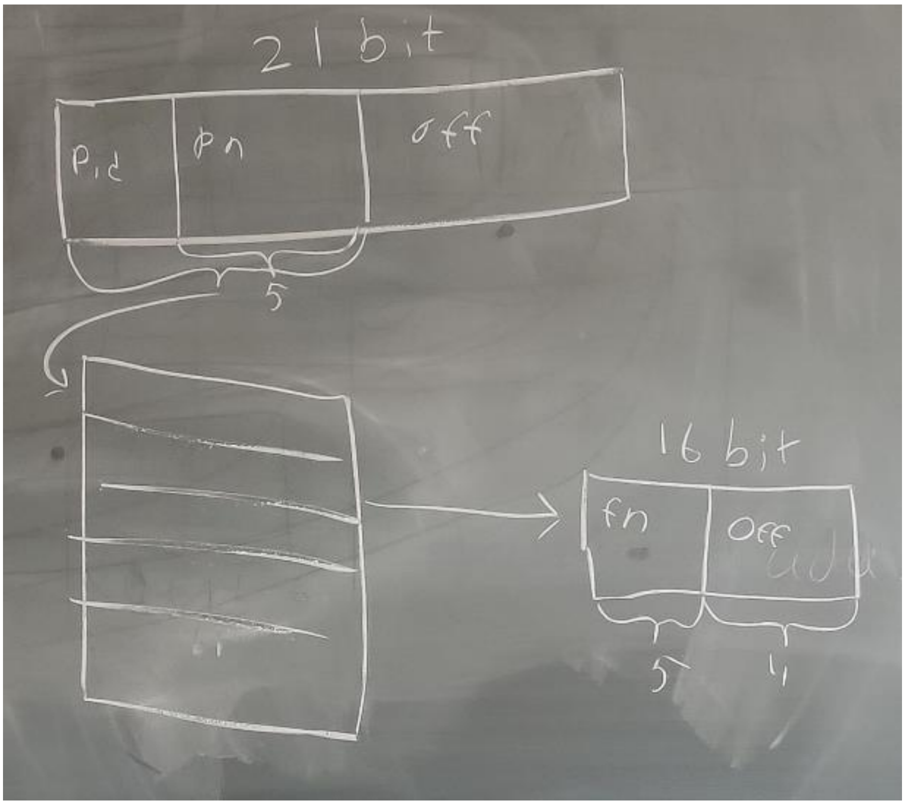
d. 650000

e. 2000001



page size = 1 KB = 1024 B

Page number	Offset
3085/1024 = 3	3085 mod 1024 = 13
42095/1024 = 41	42095 mod 1024 = 111
215201/1024 = 210	215201 mod 1024 =161
650000/1024 = 634	650000 mod 1024 =784
2000001/1024 = 1953	2000001 mod 1024 = 129



16 bit physical address, 21 bit virtual address. The inverted fragmentation table has 2^5 entries, because it has as many entries as the physical address holds, and the conventional single page table has as many as the virtual address.

Virtual Memory

(͡° ͜ʖ ͡°)

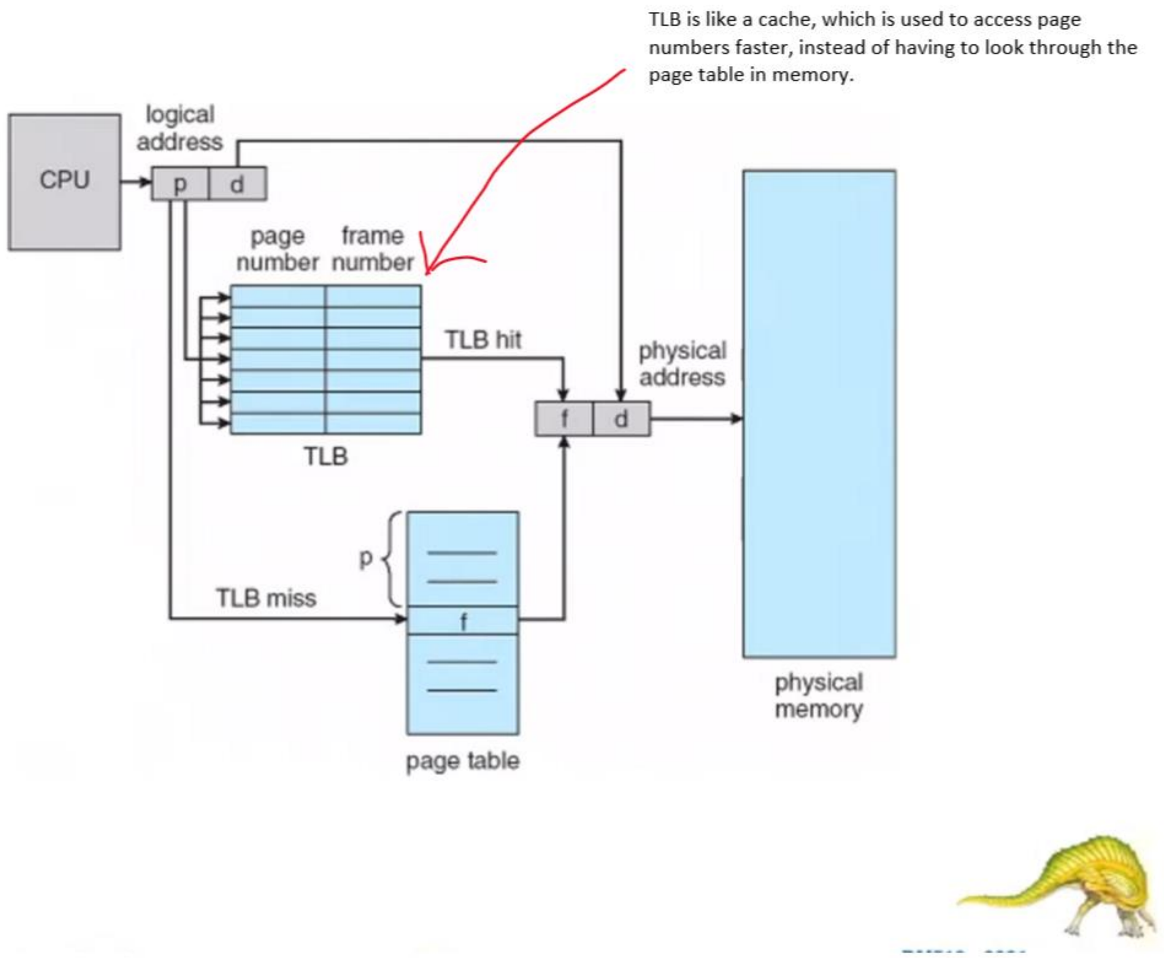
Belady’s anomaly: More frames in memory does not always give a better page-fault rate. A replacement algorithm is not optimal if Belady’s anomaly occurs.

(͡° ͜ʖ ͡°)

Demand paging: With demand-paged virtual memory, pages are loaded only when they are demanded during program execution.

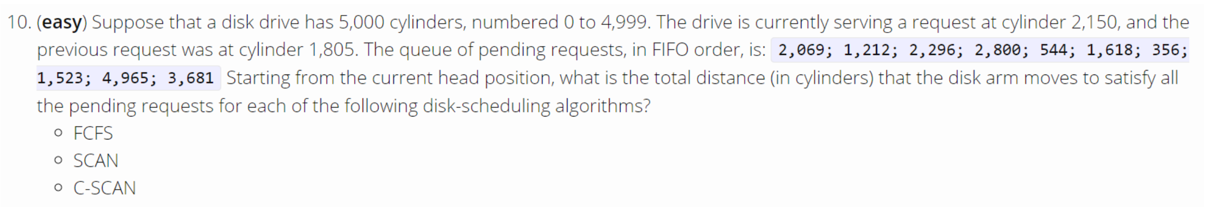
(͡° ͜ʖ ͡°)

Global vs. Local replacement: In global replacement, a process may replace a page that is owned by any process on the same CPU. In local replacement, may only replace its own.


TLB is like a cache, which is used to access page numbers faster, instead of having to look through the page table in memory.

Mass Storage

SSTF: Shortest-Seek-Time-First. An HDD I/O scheduling algorithm that sorts requests by the amount of seek time required to accomplish the request; the shortest time has the highest priority.


10. (**easy**) Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is: `2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681` Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?
   - FCFS
   - SCAN
   - C-SCAN

FCFS: Tager bare efter FIFO køen, AKA 0,1,2,3,4,5,6,7,8,9

SCAN: Scanner henover disken fra den ene ende til den anden. Siden den lige har været ved 1805 og nu er ved 2150 ved vi at den kører opad. Så de næste requests er 2,3,9,8, og så begynder den at køre nedad igen og servicer request 0,7,1,4,6.

C-SCAN: Ligesom scan, men i stedet for at køre op og ned som en elevator kører den bare videre rundt i en cirkel. Derfor er rækkefølgen 2,3,9,8,6,4,1,7,0.

MTBF: Mean-time-between-failure

Deadlocks

4 Deadlock criteria:

1. Mutual Exclusion	2. Hold and Wait
3. No Preemption	4. Circular Wait

Race condition: Appears in badly implemented concurrent code.
When the outcome depends on which process executes it's code first.
Often caused by mutual access to data, without any form of locking mechanism.

Semaphores:
- Multiple simultaneous accessors
- wait(): Decrements semaphore. If semaphore >= 0, execute the critical section. Else wait in queue
- signal(): Increments semaphore (signals critical section is done)

```
MyThreadSafeFunction(){
    wait(Semaphore)
    // Critical Section
    signal(Semaphore)
}
```

Mutex: (Mutual Exclusion):
- A object that is owned by a process
- The process locks the mutex and unlocks it when it's done

```
MyThreadSafeFunction(){
    lock(Mutex)
    // Critical Section
    unlock(Mutex)
}
```

Busy Waiting: When a process is occupying the processor, even though it is waiting. Often caused by bad implementation.
Example: Being in a while loop until a certain timestamp. Could be prevented using something like sleep().
Banker's Algorithm: Resource allocation and deadlock avoidance algorithm
Available : Indicates the number of available resources of each type.

Max : Defines the maximum demand of each process in a system.

Allocation : The number of resources of each type currently allocated to each process.

Need : Indicates the remaining resource need of each process.

- Need = Max – Allocation

Virtualization

Unikernel: (Example Mirage OS)
Single address space (No distinction between user and system)
Highly specialized for specific tasks (Example: Web API)
Can be very small in size (often < 10 MB)

PROS:

- Security: Way less code than a normal OS → Less vulnerabilities
  May not have: Bluetooth, USB stack, File sharing
- Speed: No context switches (no syscalls between user and system)
- Manageability: Less code → Less code that need updates
- Efficiency: Basically zero overhead

CONS:

- Very different from normal general purpose OSs
- Not widely used

Husk at skriv om for `--plagiat` xD

Describe the three types of traditional hypervisors?

a. Type 0—implemented by Firmware, low overhead but generally fewer features. Other VMMs can run as guests.

b.Type 1—special purpose software or general purpose operating systems that provide a means to run guests.Takes advantage of available hardware assistance, most feature rich.

c.Type 2—application providing guest execution unbeknownst to the operating system. More overhead and fewer features than type 1.

Describe four benefits of virtualization

a.Consolidating multiple physical systems into guests on Fewerphysical systems - cost, power, cooling savings.

b.Easier management - X guest machines easier to monitor, admin-ister than X physical systems.

c.Where available, live migration oF guest between systems decreases downtime, eases administration, and allows better resource management.

d.Development and QA - engineers have many operating systems and operating system versions available on a single system for development and testing