

# OPERATING SYSTEM



Memory management in your **FreeBSD**

Student name	ID
Mahmoud Essam Fathy	20221460231
Abdelrahman Ashraf Ragab	20221374041
Abdullah Hussein Ibrahim	20221427861
Zyad Ashraf Hafez	20221374025
Marwan Ali Abd-Elsatar	20221460240

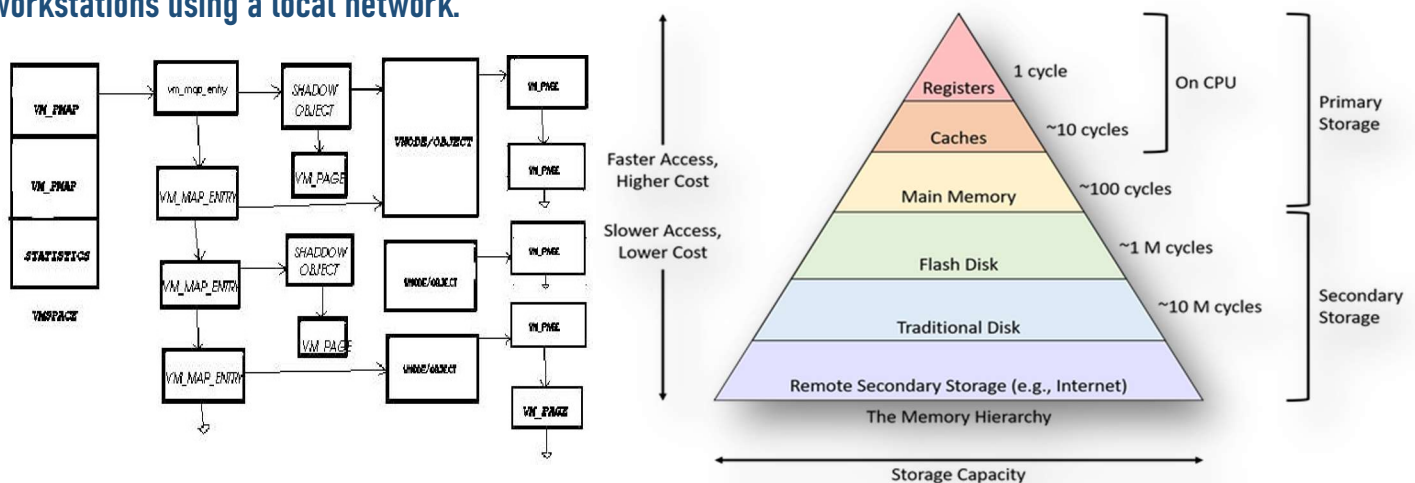
Supervised by

Dr. Yasser Fouad

## What's Memory Management?

We can consider memory as the **core** of the Personal Computer's life, as the memory management system holds a main role in every operating system, tasked with the efficient management of memory resources, organized in a **hierarchical manner**. This hierarchy usually encompasses multiple memory levels, with memory access times being inversely related to their proximity to the CPU.

The **main memory** is like the top level, and just below it is **secondary storage**, which is usually on disk drives. Some work setups use a **three-level system** with file servers or network storage connected to workstations using a local network.



## What About FreeBSD, how does it work?

- The virtual machine (VM) used by FreeBSD uses memory as **pages**, which are typically **4KB** in size on most systems.
  - Which one Page means refers to the fundamental unit of memory management within the operating system.
  - page is a **fixed-size block of memory** used for various purposes, including the allocation of physical memory and the management of virtual memory
- FreeBSD manages pageable memory using three queues. **Top(1)** displays the sizes of the three queues **Laundry, Inactive, and Active**.



## Assignment: Memory management in your FreeBSD

Subject: Operating System

Dr. Yasser Fouad

```
Oct 30 21:22
Terminal
Last pid: 1073; load averages: 0.11, 0.14, 0.05; battery: 60% up 0+00:03:07 21:22:25
77 processes: 1 running, 75 sleeping, 1 stopped
CPU: 1.2% user, 0.0% nice, 0.9% system, 0.0% interrupt, 97.8% idle
Mem: 325M Active, 240M Inact, 976K Laundry, 352M Wired, 229M Buf, 3021M Free
Swap: 1024M Total, 1024M Free

PID  USERNAME  THR  PRI  NICE  SIZE  RES  STATE  C  TIME  WCPU  COMMAND
993  mahmoud   19   21    0   3057M  449M  select  3  0:09  3.24%  gnome-shell
878  root       3    21    0   216M   98M   select  1  0:04  3.17%  Xorg
1058 mahmoud    5    20    0   125M   47M   select  2  0:01  0.89%  gnome-terminal-serv
735  root       1    20    0   13M    2364K select  2  0:00  0.11%  moused
1073 mahmoud    1    20    0   14M   3796K CPU2   2  0:00  0.09%  top
1024 mahmoud    5    20    0   67M   31M   select  2  0:00  0.01%  gsd-xsettings
982  mahmoud    5    20    0   53M   14M   select  0  0:00  0.01%  gvfs-udisks2-volume
999  mahmoud    2    20    0   609M  10M   select  1  0:00  0.01%  pulseaudio
1025 mahmoud    5    20    0   41M   9708K select  3  0:00  0.00%  gsd-housekeeping
1036 mahmoud    7    20    0   250M  77M   select  1  0:01  0.00%  evolution-alarm-not
974  mahmoud    1    20    0   14M   4564K select  2  0:01  0.00%  dbus-daemon
1042 mahmoud    5    20    0   65M   30M   select  2  0:00  0.00%  ibus-extension-gtk3
960  mahmoud    5    52    0   87M   29M   select  2  0:00  0.00%  gnome-session-binar
1003 mahmoud    5    25    0   172M  53M   select  2  0:00  0.00%  goa-daemon
776  messagebus 1    20    0   14M   4152K select  3  0:00  0.00%  dbus-daemon
1018 mahmoud   10   52    0   129M  48M   select  0  0:00  0.00%  evolution-calendar-
1016 mahmoud    5    20    0   328M  32M   select  2  0:00  0.00%  gsd-media-keys
873  root      16    20    0   88M   8864K select  3  0:00  0.00%  console-kit-daemon
1026 mahmoud    5    20    0   72M   31M   select  1  0:00  0.00%  gsd-power
1012 mahmoud    4    52    0   55M   15M   select  2  0:00  0.00%  ibus-daemon
1044 mahmoud    4    20    0   57M   28M   select  0  0:00  0.00%  ibus-x11
1010 mahmoud    5    20    0   65M   28M   select  1  0:00  0.00%  gsd-keyboard
935  root       7    20    0   94M   15M   select  2  0:00  0.00%  bsd disks
1053 mahmoud    7    52    0   150M  50M   select  1  0:00  0.00%  evolution-addressbo
1031 mahmoud    8    20    0   2174M 39M   select  1  0:00  0.00%  gjs-console
1001 mahmoud    5    23    0   96M   45M   select  0  0:00  0.00%  evolution-source-re
1005 mahmoud    8    20    0   2169M 39M   select  0  0:00  0.00%  gjs-console
1075 polkitd    4    20    0   37M   9108K select  2  0:00  0.00%  polkitd
735  mahmoud    8    26    0   82M   28M   select  0  0:00  0.00%  zeitgeist-datahub
```

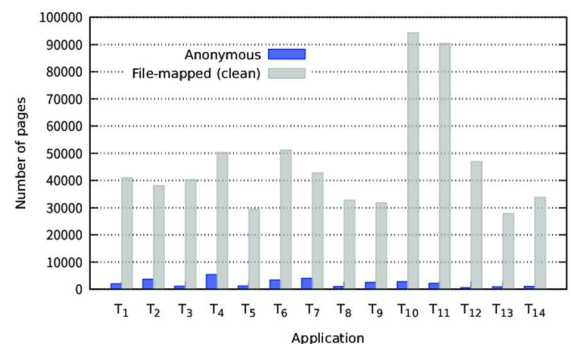
(This image show processes on the system and all needed consuming resources, even in CPU and Ram)

**Pageable memory** is like a big notebook for your computer. When your computer needs to use something, it wrote in the notebook but doesn't have enough space, it's like moving that info to another notebook (the **swap device**) and erasing it from the big one.

if your computer wants to use that info again, it checks the swap notebook, finds what it needs, and puts it back in the **big notebook**. If there's not enough space in the big notebook, it might erase some less important stuff to make room. The stuff in the swap notebook is like secret notes your computer keeps, and it's called **"anonymous memory."**

### Examples of anonymous memory include:

1. Memory allocated by malloc() in applications.
2. Contents of a swap-type MD device.
3. Data in tmpfs filesystems.
4. Shared memory in SysV or POSIX segments.

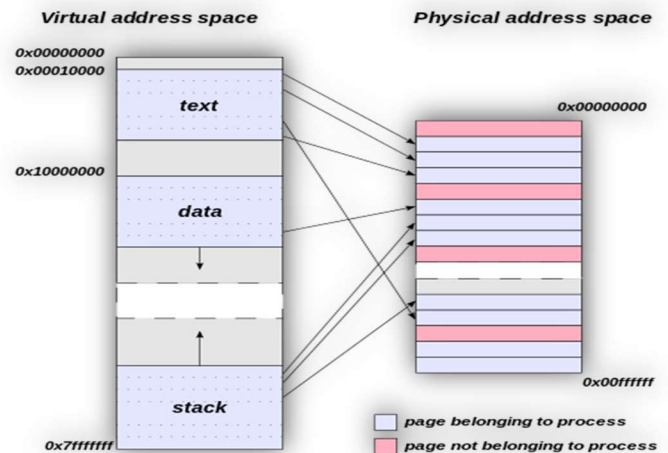
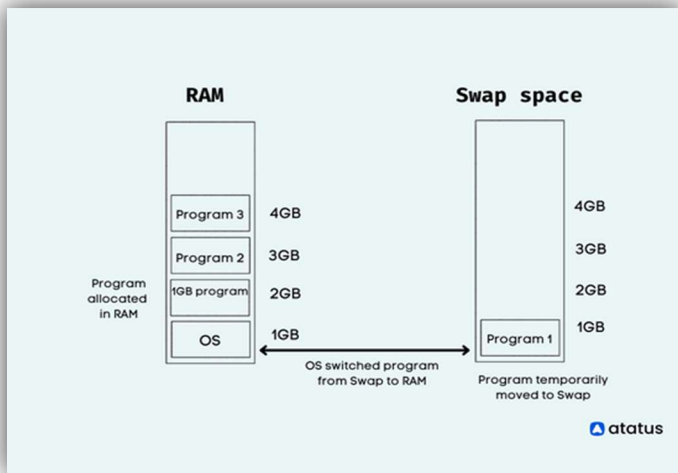


Think of file data as your computer's saved files and information. The computer has a special **memory space** just for these files. Different systems call it by different names, like "buffer cache" in **UFS**, **msdosfs**, and **NFS**, and "ARC" in **ZFS**.

When we need space for **new stuff**, the old data in the memory space **waits** in a line (like a queue) called the **"inactive queue."** But in ZFS, it's like the old data goes away right away without waiting in line.

**Pages in page queues** can be either "**clean**" or "**dirty**." Dirty pages need to be saved before they can be used again, at which point they become clean.

**Dirty anonymous** pages are cleaned by saving their contents to the **swap device**. **Dirty file pages** are cleaned by saving their contents to the **filesystem's main storage**. Once a page is clean, it's ready to be freed up and used again.



## What about memory classes?

**FreeBSD memory Classes** organized into different classes:

### Active:

- Holds **recently used pages** by programs.
- Contains both **clean and dirty pages** (clean means unaltered, dirty means changed).
- Pages are regularly checked by the page **daemon (a background process)** to see if they've been used recently.
- If a page hasn't been used in a while, it's moved to **the inactive queue**.
- It uses a **pseudo-Least Recently Used (LRU)** method to manage pages.

### Inactive:

- Holds pages that have been **moved out of the active queue**.
- Also contains pages kicked out of the **buffer cache (where files are stored temporarily)**.
- Pages are scanned when there's a **memory shortage**.
- Referenced pages go back to the **active queue**.
- Dirty pages are moved to a queue for cleaning.
- Unused, clean pages can be freed up right away.
- Uses a **second-chance LRU** method to manage pages.

PLRU is like a smarter version of LRU. It doesn't keep track of exact ages; it guesses which items to replace based on rough estimates of their age.

a combination of using a queue, similar to FIFO

## Laundry:

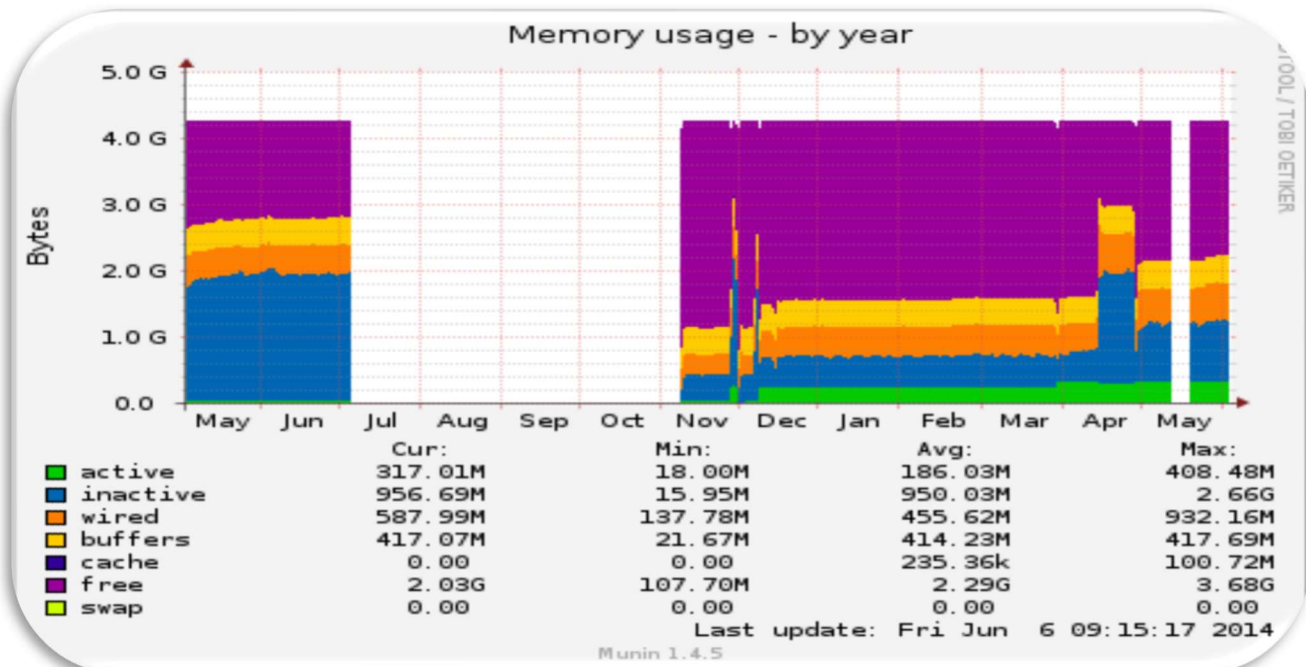
- A queue for **managing dirty (changed)** inactive pages that need cleaning before they can be used again.
- Managed by a separate thread called the **laundry thread**.
- Cleaning frequency depends on various factors.
- Referenced **pages go back to the active queue**.
- Dirty pages are cleaned and put **back in the inactive queue**.
- Helps maintain a balance between the **inactive and laundry queues**.

## Free:

- Memory that's available for use by the whole system.

## Wired:

- Non-pageable memory, meaning it can't be freed automatically.
- Userland memory can be "wired" by certain commands (**like mlock**).
- Kernel memory and the contents of the ARC and buffer cache are also wired.
- Some memory, like the kernel itself, is always wired and never released.



Main Reference Was

FreeBSD Wiki Memory, The Design and Implementation of the FreeBSD Operating System (2nd Edition)