Lecture 07

**RPM-SWAP**

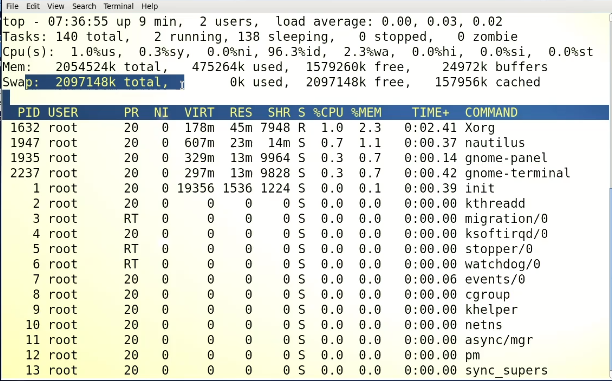
There 3 ways to install packages in Linux CentOS

1. $ rpm (manual way)
   1. $ yum (automatic way)
2. Source code installation --> from tar ball
3. SRPM --> absolute or not necessary

* .exe files are Windows installation file which can be installed on Linux with “Wine” or “crossover”
* Yum --> Yellow-dog Modifier

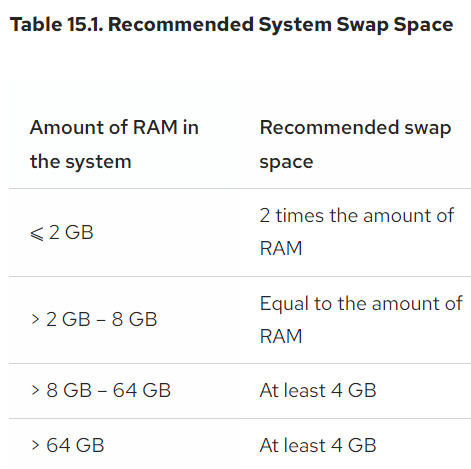
**SWAP**

- in TOP command there is a field called “SWAP”



What is SWAP (Interview Question)

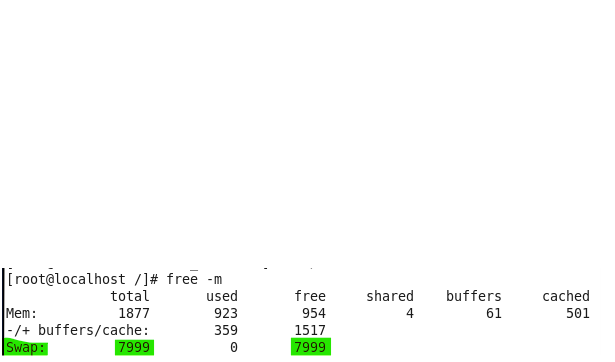
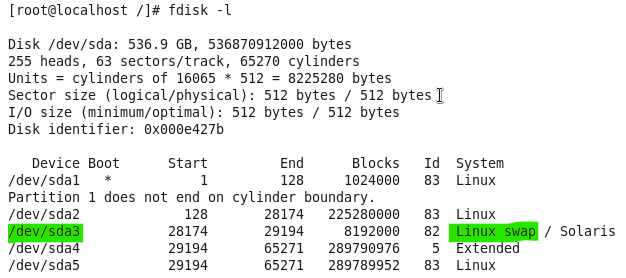
* SWAP is virtual memory
* It is a partition in HDD
* It is partial expansion of RAM
* In Linux, SWAP is a space on a hard drive that is used as virtual memory. When the system runs out of physical RAM, it temporarily moves inactive data from RAM to the SWAP space, allowing the system to continue running without slowing down. The Linux operating system automatically manages SWAP space, determining when to move data to and from the SWAP space. It is possible to add or remove SWAP space by modifying the configuration of the system.
* It works like RAM
* Is SWAP mandatory?
* No.
* SWAP uses Virtual File System --> SWAP file system --> VFS
* It only stores “processes”
* Everything goes to CPU for resources after loading in RAM 🡪 first in RAM 🡪 then in CPU
* For making “SWAP” its size should be double as compared to RAM 🡪 i.e 128 RAM 🡪 256 Mb SWAP
* **How SWAP works**
  + Suppose 128 Mb RAM can handle 2 processes at a time.
  + More than 2 processes can’t run until one process is terminated in the RAM to give the required portion of RAM to the process in queue.
  + Suppose SWAP is created 🡪
  + When a user opened 3rd application, 1 process which is existing in RAM will move to SWAP.
  + If another application is opened 🡪 another application will move to SWAP 🡪 and it leaves spaces for coming application.
  + It is also called “partial RAM”.
  + SWAP would be in use after RAM is filled.
  + OS name it “swapspace”.
  + In some application SWAP remains in use 🡪 it is requirement of some applications 🡪 therefore, SWAP must be created.
  + 1st the process shifts to SWAP, 2nd then the process in Swap takes its place in RAM
  + SWAP (double of RAM) = RAM 🡪 it’s not a thumb rule
  + Minimum recommendation of AWAP 🡪 consult RedHat web site 🡪 there would be a table. (https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/7/html/storage\_administration\_guide/ch-swapspace)



* It (recommendation of SWAP) could also be found in application documentation.

RAM

SAWP

* + The process in SWAP remains silent until invoked 🡪 after invocation this process shifts the RAM or swaps with the application 🡪 it goes to RAM and shifts the inactive process to SWAPS.
  + **Active application will always stay in RAM.**
  + In windows swapping is called “paging”
  + Can SWAP be extended?
  + Yes!
  + To check SWAP in machine
  + $ free -m 🡪 use “-h” for human readable display
  + 
  + To check which partition is SWAP
  + $ swapon -s
  + 
  + Another way to check it,
  + $ fdisk -l
  + 

**How to extend SWAP? (query)**

* + Create another partition.
  + And merge it with SWAP to consider both as “swap space”.
  + 3 ways
    - fdisk partition way
    - file way
    - fileway with different commands

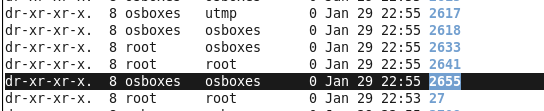
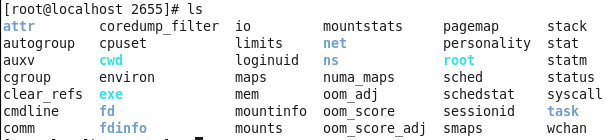
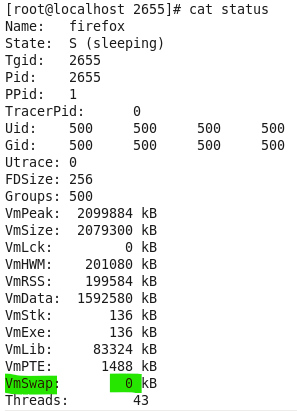
fdisk way to create and extend SWAP partition.

To extend swap space using fdisk, follow these steps:

1. Determine the current swap space: Run the **free -h** command to check the current swap space usage.
2. Create a new partition: Use the **fdisk** command to create a new partition. Example: **sudo fdisk /dev/sda**.
3. Choose the "n" option to create a new partition, then choose the "p" option for a primary partition.
4. Set the partition size by specifying the start and end sectors.
5. Choose the "t" option to change the partition type, then choose "82" for swap.
6. Choose the "w" option to write the changes and exit fdisk.
7. Format the new partition as a swap partition: Run the **sudo mkswap /dev/sda[partition number]** command. Example: **sudo mkswap /dev/sda3**.
8. Enable the new swap partition: Run the **sudo swapon /dev/sda[partition number]** command. Example: **sudo swapon /dev/sda3**.
9. Verify the new swap space: Run the **free -h** command to verify the new swap space.
10. Add entry to fstab: To persist the swap space after reboots, add an entry to **/etc/fstab** file: **sudo echo "/dev/sda[partition number] swap swap defaults 0 0" >> /etc/fstab**.

Note: Make sure to back up any important data before performing these steps, as the process involves partitioning and formatting the disk, which may result in data loss.

**To check which process is using/utilizing SWAP?**

* + Each process has a “PID””
  + /proc/ holds all the processes information in directories assigned against specific PIDs.
  + Suppose we want to check it against “firefox” browser
  + $ pidof firefox
  + 
  + $ cd /proc/ 🡪 check PID (2655) in the directory.
  + 
  + $ cd 2655
  + 
  + Check status file
  + 
  + In this case SWAP is not used by firefox browser
  + To check all processes for “swap” usage 🡪 bash scripting comes in play

|  |
| --- |
| #!/bin/bash  # Check swap usage for all processes  # Get the total swap space available  total=$(free -m | grep Swap | awk '{print $2}')  # Print header  echo "PID Process name Swap usage (MB)"  # Loop through all processes and get their PID and swap usage  for pid in $(ls /proc | grep -E '[0-9]+'); do  # Skip if the process is not a valid PID  if [ ! -d "/proc/$pid" ]; then  continue  fi  # Get the name of the process  name=$(cat "/proc/$pid/status" | grep Name | awk '{print $2}')  # Get the swap usage for the process  swap=$(grep VmSwap "/proc/$pid/status" | awk '{print $2}')  # Skip if the process is not using swap  if [ -z "$swap" ]; then  continue  fi  # Convert swap usage from KB to MB  swap=$(echo "$swap / 1024" | bc)  # Print the PID, process name, and swap usage  echo "$pid $name $swap"  done  # Print the total swap space available  echo "Total swap space: $total MB" |

* + Interview question
  + High server utilization -🡪 what to do ???