RAMDISK and SWAP TUTORIAL

1) RAM is quite fast – if one know what is doing, he or she should be able to fit in RAM memory with whole system, programs and so on.

2) RAMDISK is temporal – in case of power supply shutdown all data from current session will be lost (cheap UPS for oversensitive users will be a great solution!). These configurations are robust for restart and intended shutdown. It is auto mounted at startup.

3) RAMDISK 1.5GB – create folder:

```
    # mkdir /home/XXXXX/RAMDISK
    # sudo mount -t tmpfs -o size=1536M,mode=777 tmpfs /home/XXXXX/RAMDISK
```

4) add mount at startup append to file:

```
    #sudo vim /etc/fstab
        #   tmpfs /home/XXXXXX/RAMDISK tmpfs rw,size=1536M,mode=777 0   0
```

5) RAMDISK flush on shutdown/on restart folder

```
    # mkdir  /home/XXXXXX/RAMFLUSHDISK
    # sudo chmod 777 RAMFLUSHDISK
```

6) script: /home/XXXXXX/flushRAMDISK.sh – it is copying RAMDISK folder RAM content on HDD to adjacent folder (force, recursive and updating copy):

```
    # #!/bin/sh
    # #cp -ruf /home/XXXXXX/RAMDISK/* /home/XXXXXX/RAMFLUSHDISK/
    # #exit 0
```

7) add script execution at restart and shutdown of system:

7a) shutdown:

```
        # sudo cp flushRAMDISK.sh /etc/rc0.d/K99_flushRAMDISK
        # sudo chmod 777 /etc/rc0.d/K99_flushRAMDISK
        # sudo chmod +x /etc/rc0.d/K99_flushRAMDISK
```

7b) restart:

```
        # sudo cp flushRAMDISK.sh /etc/rc6.d/K99_flushRAMDISK
        # sudo chmod 777 /etc/rc6.d/K99_flushRAMDISK
        # sudo chmod +x /etc/rc6.d/K99_flushRAMDISK
```

SWAP DISK ARRAY:

1) there are examples of ultra-high RAM amounts demands. It could be easily resolved with hard drive RAM memory extension called SWAP. Consider making a disk array in such scenario:

-multiple hardware SSD (smaller are faster in terms of kIOps/GB) RAID 0 controllers, or multiple M.2 disks in HBA’s (PCIe adaptor)

-single software RAID 0 on above,

please refer to LINUX mdadm manpages.

2a) graphical interface: create swap area on mdadm RAID 00, obtain its UUID (LINUX disks),

```
    #sudo blkid
```

3a) configure mount point at system startup:

```
    #sudo vi /etc/fstab
        /dev/sdXX swap swap defaults 0 0
    #reboot
```

2b) terminal on local disks: swap area format and mount point

```
    #mkswap /dev/sdXX
    #sudo vi /etc/fstab
        UUID=obtainedNumber none swap sw
    #sudo mount -a
```

4) check if it is mounted:

```
    #htop
```

5) delete old swap file from system

SWAP FILE:

1) create swap file, set size and permissions

```
	#sudo dd if=/dev/zero of=/opt/swapfile.swap bs=1G count=8
	#sudo chown root:root /opt/swapfile.swap
	#sudo chmod 0600 /opt/swapfile.swap
	#sudo mkswap /opt/swapfile.swap
```

2)set mount point

```
	#sudo vi /etc/fstaw
		#/opt/swapfile.swap none swap sw 0 0
	#reboot
```

Post Scriptum: RAMDISK is not an ideal solution, like a whole system in-RAM distro. Most popular solution is to start an operating system from liveCD (at boot-up push down Tab and add ”toram” instead of „splash quiet”). In case of older hardware (< 1GB RAM) obtaining dedicated lightweight distribution (f.e. Puppy LINUX) would be better idea, than full editions like Ubuntu Server/Desktop 64bit.

Post Post Scriptum: **There are some other (slow) tricks like mounting remote Samba file server, using Dropbox, using git repositories (saving work redundantly) for obtaining additional disk space.** Pernamently one should make SSD RAID 0 (up to 4-8 harddrives) or SSD RAID 50 (RAID 0 on 2-3xRAID5; up to 8-12 harddrives) disk array as solution for saving data. With composition of Samba server; InfiniBand IB card (few computers) or 10GbE card + 10GbE switch hub (computation cluster) it should provide data handling solution. Note that there are non-network computation interconnections solutions (saving whole data locally and synchronizing changes only):

```
rsync -u -v --progress -e ssh /home/user/pathToSourceFolder/* user@IPAddr:/home/user/pathToDestinationFolder
```