

/sbin:

/sbin یک زیرشاخه استاندارد از دایرکتوری **root** در لینوکس و سایر سیستم عامل های مشابه یونیکس است که شامل برنامه های اجرایی (یعنی آماده اجرا) می باشد. آنها عمدتاً ابزارهای مدیریتی هستند که باید فقط برای **root** (یعنی administrative) در دسترس باشند.

/sbin شبیه به **/bin** است که شامل برنامه های اجرایی مورد نیاز برای راه اندازی (به عنوان مثال شروع) سیستم است، با این تفاوت که برنامه های **/sbin** معمولاً فقط توسط کاربر **root** اجرا می شوند. بنابراین، **/sbin** به طور پیش فرض در متغیر محیطی **PATH** کاربران عادی نیست، بلکه برای حساب **root** است. **PATH**، تمام دایرکتوری هایی را که سیستم برای برنامه های اجرایی جستجو می کند لیست می کند، بنابراین نیازی به وارد کردن نام مسیر مطلق برنامه ها در دستورات نیست. متغیرهای محیطی دسته ای از متغیرها هستند که به **shell** می گویند که چگونه همانطور که کاربر در خط فرمان یا در اسکریپت ها (یعنی برنامه های کوتاه) کار می کند، رفتار کند.

/root

root نام کاربری یا حساب کاربری است که به طور پیش فرض به تمام دستورات و فایل های لینوکس یا سایر سیستم عامل های مشابه یونیکس دسترسی دارد. به آن اکانت ریشه، کاربر ریشه و سوپرکاربر نیز گفته می شود. **root directory** یک دایرکتوری است که همه دایرکتوری های دیگر از جمله زیر شاخه ها و فایل ها در آن قرار دارند. دایرکتوری ریشه با یک اسلش رو به جلو (/) مشخص می شود.

/root که دایرکتوری اصلی کاربر **root** است. این دایرکتوری شامل فایل های پیکربندی کاربر است. معمولاً دایرکتوری است که کاربر هنگام ورود به سیستم، خود را در آن می یابد. **/root** یک زیر شاخه از دایرکتوری **root** است، همانطور که با اسلش شروع می شود، نباید با دایرکتوری **root** اشتباه گرفته شود. دایرکتوری های **Home** برای کاربرانی غیر از **root** به طور پیش فرض در پوشه **/home** ایجاد می شوند که یکی دیگر از زیر شاخه های استاندارد دایرکتوری **root** است.

/proc:

/proc بسیار خاص است زیرا یک فایل سیستم مجازی نیز هست. گاهی اوقات از آن به عنوان یک سیستم **pseudo-file** یاد می شود. **/proc** حاوی فایل های واقعی نیست، اما اطلاعات سیستم زمان اجرا (مانند حافظه سیستم، دستگاه های نصب شده، پیکربندی سخت افزار و غیره) را شامل می شود. به همین دلیل می توان آن را به عنوان یک مرکز کنترل و اطلاعات برای هسته در نظر گرفت. در واقع، بسیاری از برنامه های کاربردی سیستم به سادگی در **/proc** فراخوانی می شوند.

برای وارد شدن یک دایرکتوری از دستور `cd` استفاده میکنیم. به این صورت که با زدن کد دستوری :

```
# cd /sbin
```

وارد دایرکتوری `sbin` می شویم

```
root@Soroush:~# cd /
bin/      etc/      lib32/    lost+found/ opt/      run/      srv/      usr/
boot/     home/     lib64/    media/      proc/     sbin/     sys/      var/
dev/      lib/      libx32/   mnt/        root/     snap/     tmp/
root@Soroush:~# cd /sbin
root@Soroush:/sbin# |
```

با باز کردن سیستم در دایرکتوری روت قرار می گیریم اما میتوان با زدن دستور :

```
# cd ~
```

می توان از دایرکتوری های دیگر وارد `root` شد.

```
root@Soroush:~# cd /sbin
root@Soroush:/sbin# cd ~
root@Soroush:~# |
```

برای وارد شدن به `proc` نیز میتوان هم همین کار را انجام داد.

```
# cd /proc
```

```
root@Soroush:/sbin# cd ~
root@Soroush:~# cd /proc
root@Soroush:/proc# |
```

برای بررسی محتوای یک دایرکتوری می توان از دستور ls استفاده کرد
به این صورت که بعد از وارد شدن به دایرکتوری مورد نظر ls می زنیم
برای sbin :

```
root@Soroush:~# cd /sbin
root@Soroush:/sbin# ls
aa-remove-unknown      dpkg-reconfigure      installkernel          mkfs.ntfs              runuser
aa-status              dumpe2fs              invoke-rc.d            mkfs.vfat              service
aa-teardown            e2fsck                ip                     mkhomedir_helper      setcap
accessdb               e2image               ip6tables              mklost+found           setvesablink
add-shell              e2label               ip6tables-apply        mkntfs                 setvtrgb
addgnupghome           e2mmpstatus           ip6tables-legacy       mkswap                 shadowconfig
addgroup               e2scrub               ip6tables-legacy-restore modinfo                 shutdown
adduser                e2scrub_all           ip6tables-nft           modprobe                start-stop-daemon
agetty                 e2undo                ip6tables-nft-restore  mount.drivfs            sudo_logsrvd
apparmor_parser        e4crypt               ip6tables-nft-save     mount.fuse              sudo_sendlog
apparmor_status        e4defrag              ip6tables-restore      mount.fuse3             sulogin
applygnupgdefaults     ebtables              ip6tables-restore-translate mount.lowntfs-3g        swaplabel
arpd                   ebtables-nft          ip6tables-restore-translate mount.ntfs               swapon
arpables               ebtables-nft-restore  ip6tables-save         mount.ntfs-3g           switch_root
arpables-nft           ebtables-nft-save     iptables               newusers                sysctl
arpables-nft-restore   ebtables-restore      iptables-apply         nfsd                    tarcat
arpables-nft-save      ebtables-save         iptables-legacy        nft                     tc
arpables-restore       faillock              iptables-legacy-restore nologin                 telinit
arpables-save          fatlabel              iptables-legacy-save   ntfsclone               tftp
badblocks              filefrag              iptables-nft            ntfsresize              tune2fs
biosdecode             findfs                iptables-nft-restore   ntfslabel               tzconfig
blkdeactivate          fsck                  iptables-nft-save      ntfsresize              ufw
blkdiscard             fsck.cramfs           iptables-restore       ntfsundelete            unix_chkpwd
blkid                  fsck.ext2             iptables-restore-translate on_ac_power             unix_update
blkzone
```

برای root :

```
root@Soroush:/sbin# cd ~
root@Soroush:~#
root@Soroush:~# ls
desktop
```

برای proc داریم :

```
root@Soroush:/proc#
root@Soroush:/proc#
root@Soroush:/proc# ls
1  acpi          consoles      driver         ioports        kmsg           mdstat         net            stat           tty
10 buddyinfo    cpuinfo       execdomains    irq            kpagecgroup   meminfo       pagetypeinfo  swaps          uptime
4  bus           crypto        filesystems    kallsyms       kpagecount    misc          partitions    sys            version
8  cgroups      devices       fs             kcore          kpageflags    modules       schedstat     sysvipc        vmallocinfo
9  cmdline     diskstats    interrupts     key-users      loadavg        mounts        self          thread-self    vmstat
90 config.gz    dma          iomem         keys           locks          mtrr          softirqs      timer_list     zoneinfo
```

سوال دوم از تمرین یک

فایل سیستم (File System) رابطی است میان «سیستم عامل» و «سخت افزار» و وظیفه‌ی مشخص کردن مکان هر فایل و سازماندهی فایل‌ها را بر روی سخت افزار را دارد. هر دو سیستم عامل ویندوز و لینوکس فایل‌ها را بر اساس دیسک در هرمی از دایرکتوری‌ها قرار می‌دهند. این دایرکتوری‌ها معمولاً در GUI فولدر یا پوشه نامیده می‌شوند و کل هرم نیز فایل سیستم نامیده می‌شود. معماری ویندوز و لینوکس در برخی جاها یکسان است، اما در مورد ابزارها تفاوت بسیاری دارند.

در ویندوز تنها یک حرف برای هر فایل سیستم وجود دارد، مانند C یا D. در لینوکس هیچ حرفی برای درایو وجود ندارد، پس تنها یک فایل سیستم روی / سوار شده است و بقیه فایل سیستم‌ها روی subdirectory ها یا زیر آدرس‌های / سوار شده‌اند.

ویندوز از FORMAT.EXE برای فرمت کردن دیسک استفاده می‌کند، در حالی که لینوکس از mkfs make file system در حالت‌های مختلف استفاده می‌کند.

هر فایل سیستم در ویندوز دارای جدول تخصیص فایل است مانند FAT، VFAT و ... که نشان می‌دهد کدام بلاک دیسک بالاترین دایرکتوری را نگه می‌دارد. در لینوکس مورد مشابه به عهده سوپر بلاک (superblock) است. یک فایل سیستم لینوکس چندین کپی از سوپر بلاک را در دیسک نگهداری می‌کند.

چند مورد از فایل سیستم‌ها در ویندوز :

FAT

FAT 32

NTFS

چند مورد از فایل سیستم‌ها در لینوکس :

Ext

Ext2

Ext3

Ext

JFS

XFS

همچنین لینوکس دارای فایل سیستم‌های msdos و vfat است که برای سازگاری با ویندوز و DOS است.