

10. hafta 1. ders;

----- tee komutu

çıktı bir dosyaya, ekrana ya da pipe a aktarılabilir.

işlemi > gibidir.

date | tee now -> data içeriği now içine aktarılır.

ls | tee list | wc -> ls çıktısının wc içeriği elde edilir.

ps -ael | tee processes | grep "\$UID" -> çalışan prosesler prosese aktarılır. \$UID aranır.

herhangi bir değişkenin değeri de aranabilir.

-a -> başkalarına ait olan prosesleri gösterir

-e = -A -> bütün prosesler seçilir.

-l -> prosesleri listeler.

Dosyaları dosya tanımlayıcılarıyla ilişkilendirme **Associating Files with a File Descriptor**

standart giriş (STDIN), 0 dışardan giriş alınır.

standart çıkış (STDOUT), 1 terminale çıktı verir.

standart Error (STDERR), 2 terminale çıktı verir.

her komut için 3 tane dosya işaretçisi vardır. Sistem tarafından tanımlıdır.

bu dosyalarla ilişkilidir ama isteğe bağlı olarak bir dosyaya aktarılabilir.

yazma modunda bir dosya açmak için;

- `exec n>file`
- `exec n>>file`

okuma modunda açmak için;

- `exec n<file`
- `exec n<<file`

hem yazma hem de okuma modunda açmak için;

- `exec n<>file`
- `exec n<<>>file`

Çalışma şekli

1. dosya varsa dosyanı içeriğini temizler.
2. o dosya ekleme modunda açılır.

n değerleri;

n bir integer değerdir.

n = 0 ise giriş | okuma işlemi yapar. verilen dosyanın içeriğini okur.

n=1 ise çıktı | standart çıktı verilen dosyaya yönlendirilir.

n=2 ise error

exec 5>zor // dosya yoksa oluşturur. varsa içeriğini temizler.

ps aux 1>&5 // normal çıktıyı yönlendirir. 5 pointerına yani zor dosyasına. &5 yerine zor yazılabilir.

date 1>>zor // date içeriği zora eklenir. 1 olmasa da olur.

exec 6<file // file ile 6 ilişkilendirilmiştir. okuma modunda

exec 7<zor // zor okuma modunda 7 ile ilişkilendirilir.

cat 0<&6 // 6 ile ilişkilendirilen dosyayı cat ile oku

cat 0<&7 // 7 yi oku

if [-f zor] // zor dosya olarak var mı

i=0 // var i= 0

while read LINE // satırı oku

do

i = `echo "\$i + 1" | bc` // i bir artar. ekrana yazar. pipe ile bc komutuna gönderilir.

done<zor // bu dosya üzerinden okuma işlemi yapılacak. read ile okunup LINE a aktarılacak

echo \$i // i değeri döner yani satır sayısı bulunur.

fi

bc = exact içine aktarılan komutta matematik işlemi varsa onu yapar.

```
dizincarcala x path x centos x processes x dosya-satir-sayisi x file x
1 #!/bin/bash
2 exec 5>zor
3 ps aux 1>&5
4 date 1>>zor # date >> zor
5 exec 6<file
6 exec 7<zor
7 cat 0<&6 # cat <&6
8 cat 0<&7 # cat <&7
9 if [ -f zor ]
10 then
11     i=0
12     while read LINE
13     do
14         i=`echo "$i + 1" | bc`
15     done < zor
16     echo $i
17 fi
18
```

```
neo@ubuntu:~$ echo selan
selan
neo@ubuntu:~$ echo selan > processes
neo@ubuntu:~$ clear
neo@ubuntu:~$ bash dosya-satir-sayisi
deneme-
deneme2-
deneme3-
deneme3.cpp-
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0  0.1 33892 4324 ?        Ss   Dec20   0:01 /sbin/init
root         2   0.0  0.0   0     0 ?        S    Dec20   0:00 [kthreadd]
root         3   0.0  0.0   0     0 ?        S    Dec20   0:00 [ksoftirqd/0]
root         5   0.0  0.0   0     0 ?        S<   Dec20   0:00 [kworker/0:0H]
root         7   0.0  0.0   0     0 ?        S    Dec20   0:43 [rcu_sched]
root         8   0.0  0.0   0     0 ?        S    Dec20   0:00 [rcu_bh]
root         9   0.0  0.0   0     0 ?        S    Dec20   0:00 [migration/0]
root        10   0.0  0.0   0     0 ?        S    Dec20   0:01 [watchdog/0]
root        11   0.0  0.0   0     0 ?        S    Dec20   0:01 [watchdog/1]
root        12   0.0  0.0   0     0 ?        S    Dec20   0:00 [migration/1]
root        13   0.0  0.0   0     0 ?        S    Dec20   0:00 [ksoftirqd/1]
root        15   0.0  0.0   0     0 ?        S<   Dec20   0:00 [kworker/1:0H]
root        16   0.0  0.0   0     0 ?        S    Dec20   0:00 [kdevtmpfs]
root        17   0.0  0.0   0     0 ?        S<   Dec20   0:00 [netns]
```

zor dosyasının içeriği

```
root      58623   0.0  0.0   0     0 ?        S    17:28   0:00 [kworker/0:0]
neo       58649   0.0  0.0 16624 2744 pts/0    S+   17:32   0:00 bash dosya-sat
ir-sayisi
neo       58650   0.0  0.0 22644 2676 pts/0    R+   17:32   0:00 ps aux
Tue Dec 22 17:32:59 MSK 2020
263
neo@ubuntu:~$
```

toplam satır sayısını gösterir.

for FILE in \$FILES

do

ln -s \$FILE ./docs >> /tmp/ln.log 2> /dev/null // s. link eklenir. hata verirse dev null'a gider.

done

10. hafta 2. ders

ln -s -> hedef directory, sadece hedef,

ls -s -> dosya s.link adı

hedef olarak

path belirtilecekse -s kaldırılır. aynı isimle s.link eklenir.

command > *file* 2>&1 -> hata ve çıktı 1 ile ilişkilendirilen dosyaya aktarılır.

----- Standart error çıktısı

echo *string* 1>&2 -> çıktı terminale gelir.

printf *format args* 1>&2 -> çıktı terminale gelir.

if [! -f \$FILE] ;

then echo "ERROR: \$FILE is not a file" >&2 ;

fi -> dosya değilse hata retür. terminale yazar.

----- Fonksiyonlar

fonksiyon_adi () {} şeklinde tanımlanır.

source mycd-> ile fonksiyon dosyası terminale tanıtılır. bağlantı kurulur.

----- örnek

lspath() { // fonk adı

OLDIFS="\$IFS" // değişkene çevre değişkeninin değeri verilir. değeri boşluktur.

IFS=: // dizinler arasında : kullanılıyor. normalde " " dur ayırma karakteri

for DIR in \$PATH ; // path çevre değişkeninden parçalama yapar.

do

echo \$DIR ; // ayrılan dizin ekrana yazdırılır. path değeri ayrılır. tek tek

done

IFS="\$OLDIFS" // sistemin değişkeni orijinal halini kaybetmesin diye eski değeri geri verilir.

}

```

1 lspath()
2 {
3     OLDIFS=$IFS
4     IFS=:
5     for DIR in $PATH ; do echo $DIR ; done
6     IFS=$OLDIFS
7 }
8
9 neo@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
10 neo@ubuntu:~$ source dizinparcala
11 neo@ubuntu:~$ lspath
12 /usr/local/sbin
13 /usr/local/bin
14 /usr/sbin
15 /usr/bin
16 /sbin
17 /bin
18 /usr/games
19 /usr/local/games
neo@ubuntu:~$

```

sonuç

lspath | grep "/usr/bin/" -> fonksiyonu koş, verilen dizini ara. önce source ile fonk terminale tanıtılmalı.

----- örnek

```

SetPath() {
    PATH=${PATH:="/sbin:/bin"}; // Path değişkenine verilen değeri atar.
    for _DIR in "$@" // dir değişkeni gelen parametreleri dizi şeklinde alır.
    do
        if [ -d "$_DIR" ]; // dir dizinde
        then PATH="$PATH": "$_DIR" ; // girilen dizini path'e aktarır. aralarına : ekler
        fi
    done
    export PATH // yeni path değeri export edilir. geçerli terminal boyunca kullanılır.
    unset _DIR // dir yok edilmiş.
}

```

```

neo@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
neo@ubuntu:~$ source dizinparcala
neo@ubuntu:~$ lspath
/usr/local/sbin
/usr/local/bin
/usr/sbin
/usr/bin
/sbin
/bin
/usr/games
/usr/local/games
neo@ubuntu:~$ SetPath /dev/block /etc/acpi /media/neo
neo@ubuntu:~$ lspath
/usr/local/sbin
/usr/local/bin
/usr/sbin
/usr/bin
/sbin
/bin
/usr/games
/usr/local/games
/dev/block
/etc/acpi
/media/neo

```

iki fonk ile path e yeni değişkenler eklenmiş oldu