

Operacijski sistemi

vaje 11

prestreganje signalov

- sintaksa: `trap akcija signal`
- `trap -l`

```
#!/bin/bash
```

```
# uлови signal CTRL-C in izvedi funkcijo sorry():
```

```
trap sorry INT
```

```
sorry() {
```

```
    echo "Tega ne morete narediti."
```

```
    sleep 3
```

```
}
```

prestreganje signalov

```
#!/bin/bash
trap '' INT
for i in `seq 1 30`; do
    echo "$i"
    sleep 1
done
```

naloga (1)

Napišite skripto `stevec.sh`, ki bo štela od 1 do n v enosekundnih intervalih (n podamo kot argument skripti).

- Skripta mora prestregati signal `SIGTERM` in v tem primeru izpisati, da program trenutno opravlja pomembno operacijo in se ne sme zaključiti.
- Kadar prejme signal `SIGUSR1`, v ozadju požene program `xeyes`.

kritični odsek

Kaj dela sledeča koda, če jo poženemo 100-krat (a) zaporedno ali pa (b) vzporedno?

```
#!/bin/bash
if [ -e number.txt ]; then
    echo "Datoteka obstaja"
else
    echo "Datoteke ni, zato ustvarimo novo"
    echo 1 > number.txt
fi

a=$(tail -n 1 number.txt)
if [ $a -gt 1 ]; then
    a=$(tail -n 1 number.txt)
    expr $a - 1 >> number.txt
fi

if [ $a -lt 2 ]; then
    a=$(tail -n 1 number.txt)
    expr $a + 1 >> number.txt
fi
```

kritični odsek

Implementirajmo zaklepanje vstopa v kritični odsek z datoteko script.lock.

```
#!/bin/bash
if [ -e number.txt ]; then
    echo "Datoteka obstaja"
else
    echo "Datoteke ni, zato ustvarimo novo"
    echo 1 > number.txt
fi
```

```
a=$(tail -n 1 number.txt)
if [ $a -gt 1 ]; then
    a=$(tail -n 1 number.txt)
    expr $a - 1 >> number.txt
fi
if [ $a -lt 2 ]; then
    a=$(tail -n 1 number.txt)
    expr $a + 1 >> number.txt
fi
```

kritični odsek

Implementirajmo zaklepanje vstopa v **kritični odsek** z datoteko `script.lock` (**preverjamo obstoj datoteke**). Ali sedaj poganjanje vzporedno deluje enako kot zaporedno? Zakaj?

```
if [ -e number.txt ]; then
    echo "Datoteka obstaja"
else
    echo "Datoteke ni, zato ustvarimo novo"
    echo 1 > number.txt
fi
while [ -e script.lock ]; do
    echo -n "."
done

touch script.lock
a=$(tail -n 1 number.txt)
if [ $a -gt 1 ]; then
    a=$(tail -n 1 number.txt)
    expr $a - 1 >> number.txt
fi
if [ $a -lt 2 ]; then
    a=$(tail -n 1 number.txt)
    expr $a + 1 >> number.txt
fi
rm -f script.lock
```

kritični odsek – `lockfile`

- implementacija z datotečnim semaforjem
- `lockfile` del paketa `procmail`

```
lockfile myscript.lock
```

```
# zacetek kriticnega odseka
```

```
...
```

```
# konec kriticnega odseka
```

```
rm -f myscript.lock
```


kritični odsek – lockfile

```
#!/bin/bash
if [ -e number.txt ]; then
    echo "Datoteka obstaja"
else
    echo "Datoteke ni, zato ustvarimo novo"
    echo 1 > number.txt
fi
```

```
lockfile script.lock
a=$(tail -n 1 number.txt)
if [ $a -gt 1 ]; then
    a=$(tail -n 1 number.txt)
    expr $a - 1 >> number.txt
fi
if [ $a -lt 2 ]; then
    a=$(tail -n 1 number.txt)
    expr $a + 1 >> number.txt
fi
rm -f script.lock
```

kritični odsek – flock

- Še ena implementacija z datotečnim semaforjem – flock.
- Sintaksa: `flock datoteka -c ukaz`

- primer:

```
function kriticniodsek {  
    # tukaj implementiramo kritični odsek  
    echo "začetek odseka"  
    read vnos  
    echo "konec odseka"  
}  
  
# izvozimo funkcijo  
export -f kriticniodsek  
read vnos  
# uporabimo datoteko semafor.lock za zaklepanje  
flock semafor.lock -c kriticniodsek  
echo "konec skripte"
```

kritični odsek – flock

- Primer uporabe flock z eksplicitnim datotečnim deskriptorjem.

```
read vnos
(
  flock 200
  # tukaj implementiramo kritični odsek
  echo "zacetek odseka"
  read vnos
  echo "konec odseka"
) 200>semafor.lock

echo "konec skripte"
```

ukazi

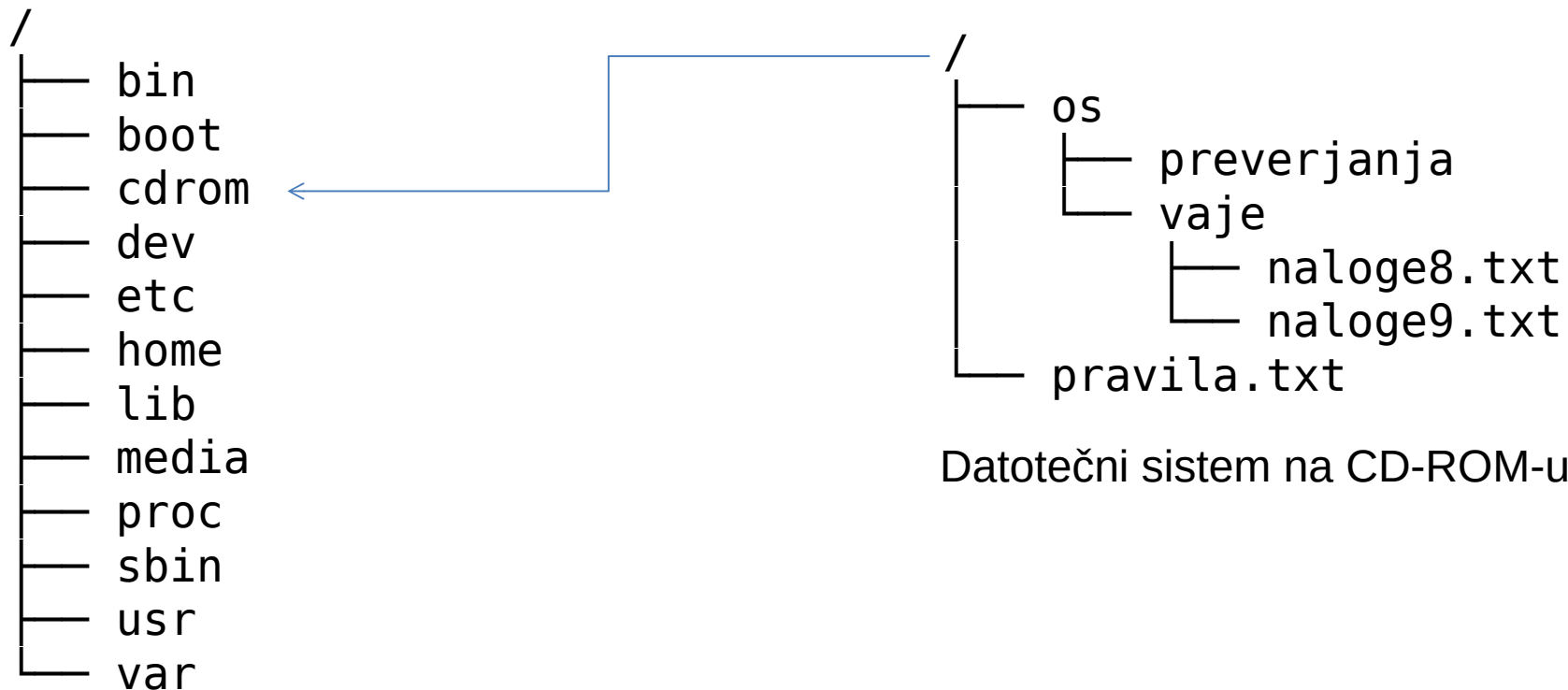
- at, atq, atrm
- cron, crontab
- mount, umount
- df, du

naloge (2)

- 1.V konzoli preverite datum in uro in ju nastavite na pravilne vrednosti, če trenutno niso pravilne.
- 2.V datoteko `ukaz.txt` napišite ukaz, ki bo na konec datoteke `status.txt` dodal besedilo
`Trenutno je: <trenuten datum in ura>`
Nato napišite ukaz, ki bo dne 20. 05. 2020 izvedel ukaz iz datoteke `ukaz.txt`.
- 3.Ukaz iz datoteke `ukaz.txt` naj se izvede čez 1 minuto in ob 4h zjutraj čez 10 dni.
- 4.Kako bi pogledali, kdaj se bodo izvedli ukazi, katerih čas še ni prišel? Kateri ukazi se bodo izvedli?
- 5.Iz vrste izbrišite ukaz, ki naj bi se izvedel ob 4h zjutraj čez 10 dni.
- 6.Nek ukaz bi radi izvajali periodično. S katerim ukazom bomo to storili?
- 7.V `crontab` dodajte vrstico z ukazom, ki bo v datoteko `status.txt` vsako minuto dopisal trenutni datum in uro.
- 8.Kako bi napisali vrstico z ukazom v `crontabu`, ki bo pognala isti ukaz kot v prejšnji nalogi vsak ponedeljek, sredo, četrtek in petek, od februarja do aprila, 5 minut čez vsako sodo uro.

Priklapljanje datotečnih sistemov

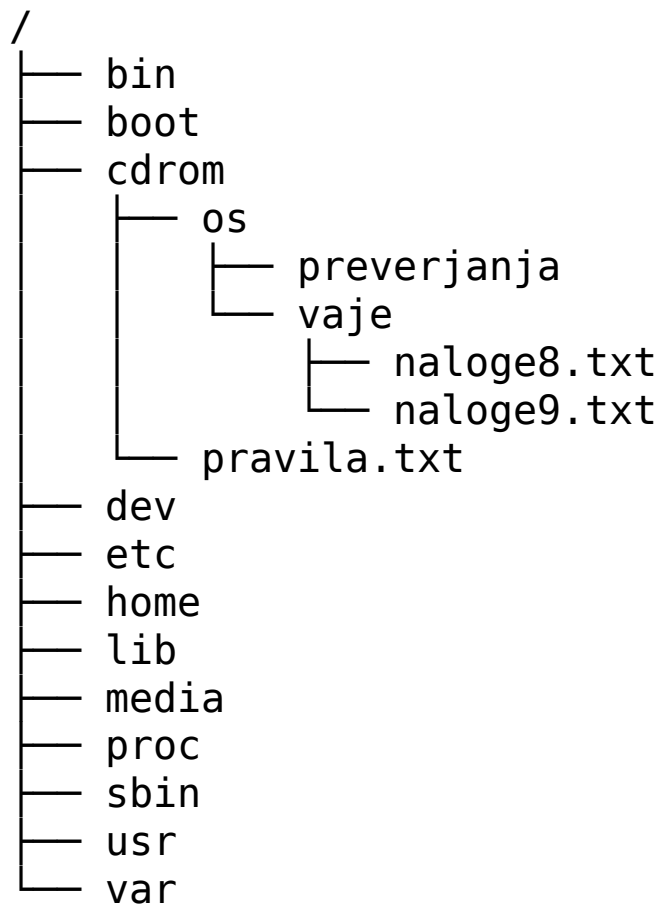
`mount /dev/cdrom /cdrom`



Datotečni sistem na CD-ROM-u

Datotečni sistem Linuxa

Priklapljanje datotečnih sistemov



naloge (3)

1. Izpišite zasedenost priklopljenih datotečnih sistemov. Izpis naj vsebuje človeku lažje berljive enote (kiB, MiB itd.).
2. Koliko prostora zavzame na disku domači imenik uporabnika `administrator` (ali pa `student`) z vsemi podimeniki?
3. Koliko prostora na disku zavzame datoteka `~/status.txt`? Kako velika je vsebina datoteke `~/status.txt`?