

Operacijski sistemi

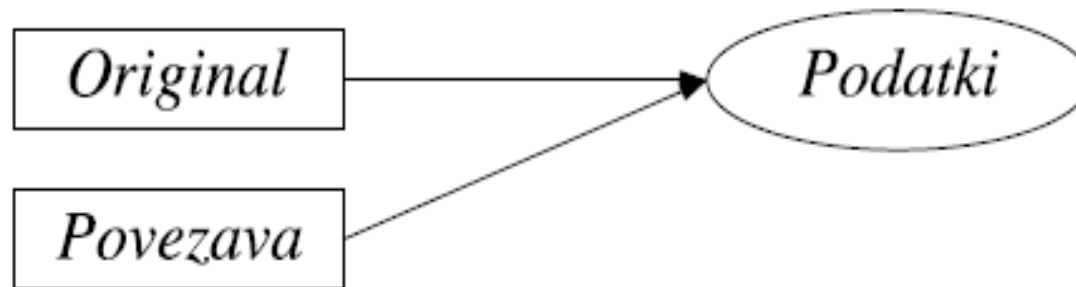
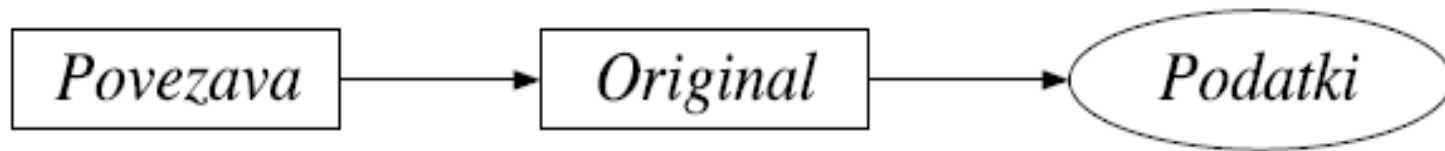
vaje 3

Tipi in Vrste datotek

Oznaka	Opis	Primer	ls --color=auto
-	navadna datoteka	/etc/passwd	bela/siva, izvršljiva (zelena)
d	imenik	/dev	modra
b	bločno orientirana naprava trdi disk	/dev/hda, /dev/sda	rumena na temno sivi
c	znakovno orientirana naprava terminal	/dev/tty	rumena na temno sivi
l	simbolična povezava navadno	/usr/src/linux, /vmlinuz	svetlo modra
p	pipa	ustvarimo z mkfifo	oker na temnosivi
s	vtičnica	mrežna povezava	vijolična

- Kako pogledamo tip in vrsto vsebine datoteke z ukazoma `ls` in `stat`?

Trde in simbolične (mehke) povezave



Naloge (1/2)

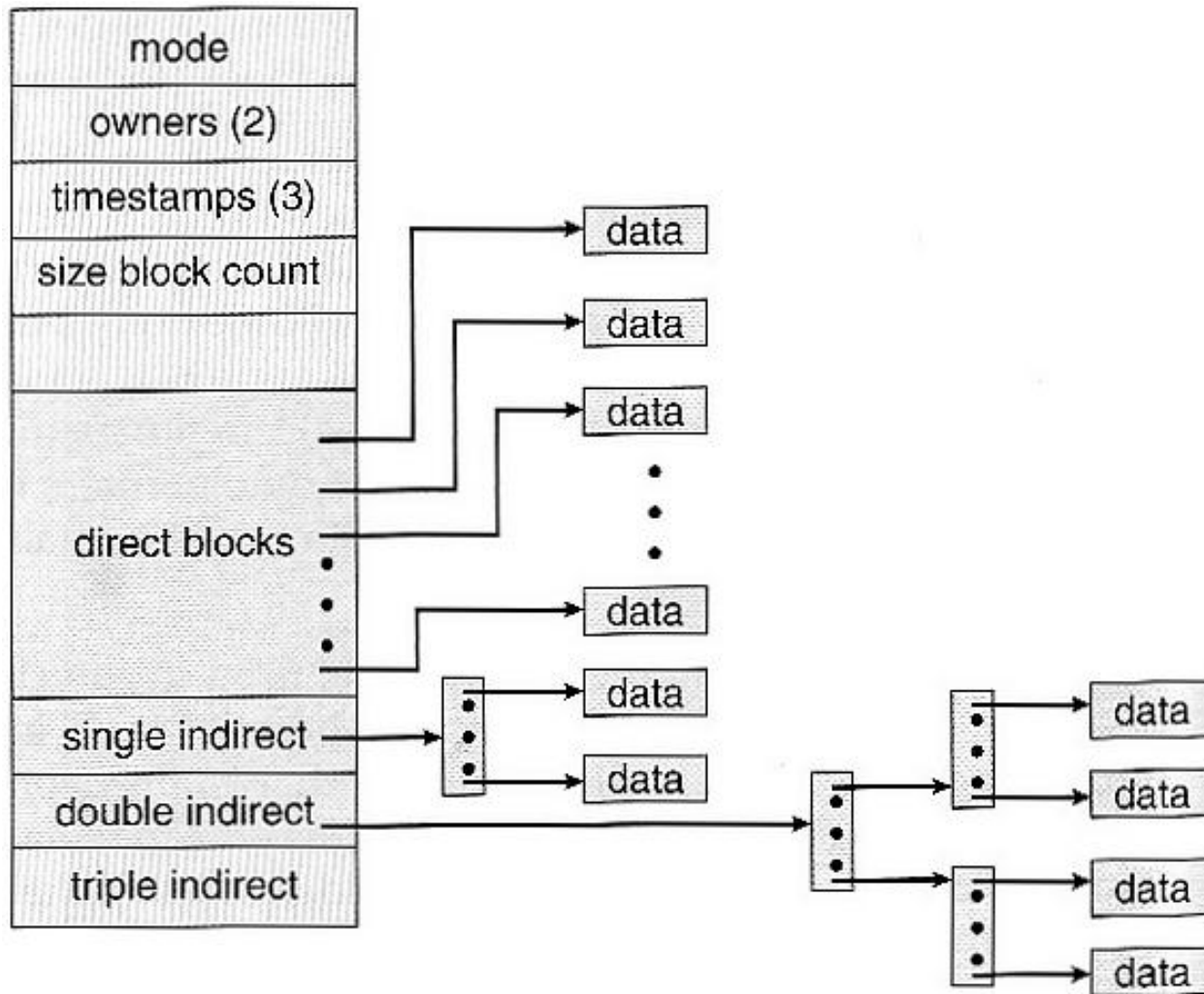
1. Napišite skripto `ustvari.sh`, ki v podanem imeniku (imenik podamo kot prvi argument) ustvari datoteke `test1.txt`, `test2.txt` in `test3.txt`!

Skripto poženite z imenikom `vaje` kot argument. Znotraj skripte so argumenti, ki smo jih podali, shranjeni v spremenljivkah `$#` (kjer `#` pomeni zaporedno številko argumenta – prvi argument npr. dobimo v spremenljivki `$1`).

V ustvarjene datoteke napišite poljubno vsebino (`gedit`, `vi`).

2. Ustvarite mehko povezavo `konfiguracije` na imenik `/etc`! Preverite, če povezava deluje! Kako to preverite?
3. Ustvarite mehko povezavo z imenom `simbolp` na `test1.txt`! Spreminjajte vsebino datoteke `test1.txt` in `simbolp`!
4. Zbrišite datoteko `test1.txt`! Preverite, kako deluje povezava.
5. Ustvarite trdo povezavo z imenom `trdap` na `test2.txt`! Spreminjajte vsebino datoteke `test2.txt` in `trdap`!
6. Zbrišite datoteko `test2.txt`! Preverite, kako deluje povezava.

inode



Naloge (1/2 - nadaljevanje)

7. Ustvarite trdo povezavo `test2.txt` na datoteko `trdap`!
8. S pomočjo ukaza `stat` pogledajte, na katere inode kažejo datoteke `test2.txt`, `test3.txt` in `trdap`!
9. S pomočjo ukaza `ls` za vsak zapis (tudi za skrite) poiščite, koliko zapisov na datotečnem sistemu kaže na isti inode.
10. Koliko prostora zavzame jedro linuxa? Kako lahko to pogledamo?

izvajanje ukazov

- zaporedno izvajanje ukazov

```
ukaz ; ukaz
```

- status ukaza

```
ukaz  
echo $?
```

- pogojno izvajanje ukazov:

```
ukaz && ukaz  
ukaz || ukaz
```

- združevanje ukazov

```
{ ukazi; }
```

- izvajanje v podlupini:

```
(ukazi)
```

- rezultat ukaza kot spremenljivka `$ (ukazi)` ali pa ``ukazi``

izvajanje ukazov

- Kako se izvajajo naslednji ukazi ob različnih vrednostih izhodnih statusov ukazov?

1. ukaz1; ukaz2

2. ukaz1 && ukaz2

3. ukaz1 || ukaz2

4. ukaz1 && ukaz2 || ukaz3

5. ukaz1 && ukaz2 && ukaz3

6. ukaz1 || ukaz2 && ukaz3

7. ukaz1 || ukaz2 || ukaz3

8. ukaz1 && { ukaz2 || ukaz3; }

Naloge (2/2)

1. Kakšen izhodni status vrne ukaz `ls ~`
2. Kakšen izhodni status vrne ukaz `ls /os`
3. Kakšen izhodni status vrne ukaz `ls -R /`, če ga prekinemo s `ctrl+c`
4. V `man ls` pogledajte, kaj pomeni posamezni izhodni status.
5. Kakšen izhodni status vrne `echo "OS"`
6. Kakšen izhodni status vrne `(ls; exit 0)`
7. Kakšen izhodni status vrne `(ls; exit 10)`
8. Kaj se zgodi, če poženemo `{ ls; exit 0; }`
9. Napišite skripto `vsebina.sh`, ki sprejme kot prvi argument imenik. Skripta naj izpiše kanonično pot do imenika (s pomočjo ukazov `readlink` ali `realpath`) in podrobnosti njegove vsebine (s pomočjo ukaza `ls`). Če se skripta zaključi uspešno, naj vrne status 0, če se zaključi neuspešno, naj vrne status 10.
10. V imeniku `~/vaje3` ustvarite datoteko `vaje3.txt` in jo premaknite v imenik `~`, če bo premik uspešen, izpišite niz "datoteka uspešno premaknjena"