

SUREČEKOVÁ

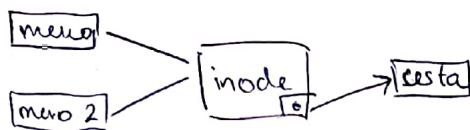
$$15 + 8 + 13 + 15 + 15 = 66$$

# 1. pevný a symbolický link

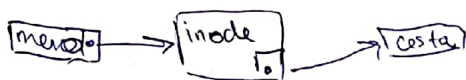
iluzeta Jurčeková

~~Pevný a symbolický link~~

Pevný - môže byť použitý iba na súbory - nie na adresáre  
tieto dva súbory poukazujú na rovnaký inode



Symbolický



typ symbolický  
link

pri premiestnení súboru inam v počítači symb. link zanikne

1. zapísal do súboru.txt hello
2. vytvoril pevný link na súbor.txt
3. vypoísal hello so súbor.txt
4. vytvoril druhý pevný link ~~ten~~ tentokrát na link1.txt
5. vypoísal ~~ten~~ obsah súbor.txt
6. vytvoril symbolický link na súbor.txt
7. symbolický link na link1.txt
8. odstranil súbor.txt
9. keďže link1.txt je pevný link aj po odstránení originálneho súboru stále ostáva
10. link 1 bol symbol. link a keď po odstránení zaniká - ukazuje chybu
11. link 2 ďalej ukazuje na link1.txt
12. odstranil link1.txt
13. link 2 je pevný link takže funguje ďalej
14. link 2 je symbolický a tak zanikol - ukazuje chybu
15. nový adresár dir
16. ~~ten~~ pevný link sa nedá urobiť pre adresár - chyba
17. vznikol symb. link na dir
18. ldir neexistuje - chyba
19. nevypíše nič - je to link na dir a ten je prázdny

15

(2.)

3  
1  
2  
0  
2  
1  
2

fork() za každým vytvorí nový child proces.

ten nič nevykonáva pretože návratová

hodnota child procesu je 0.

návratová hodnota parent procesu je  $> 0$

a odpovedá pidu dieťaťa.

Výstup tohto programu začína číslom 2

pretože pred výpisom je wait a teda

parent proces čaká na ~~zbo~~ child procesy.

Pri troch forkoch mi vznikne 7  
child procesov.

Preto máme aj 7 výpisov. ~~Parent~~

vždy čaká na ~~zbo~~ child a až potom

sa vypíše. Máme 4 (2) dve (1) a jednu (0)

pretože pri  $i=0$  vznikol je parent a 1 child

pri  $i=1$  ~~zbo~~ po dva

a pri  $i=2$  po 4

HHH. ale toto  
nezodpovedá  
poradiu, len  
počtu tých  
čísel

(3.) a) ~~exp~~ "baba" ✓

b) "a"

c) "a\$"

d)  $^ [a b]$   $^ [^ a]$

e)  $-v$  "aa"

f) ~~"^ ( . \* \ ) ' ' . \* \ 1 \$"~~  $^ \ ( . \ \backslash \ ) . \ * \ \backslash \ 1 \ \$$

g)  $^ \ ( . \ * \ \backslash \ ) ' \ ' . \ * \ \backslash \ 1 \ \$$

to je prepínač grupy, nie neg. výraz. -2b

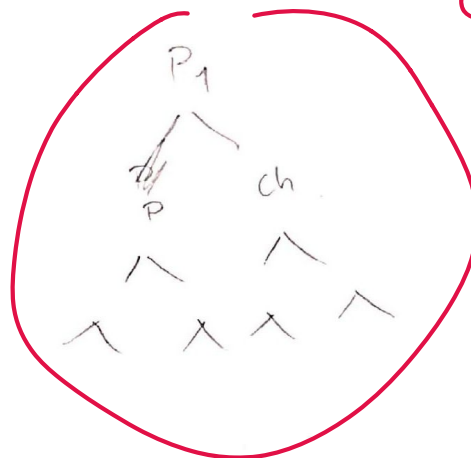
↓  
jedine dobre toto

3.

Alžbeta Jurečková

- a) ~~grep "baa"~~
- b) ~~grep "a"~~
- c) ~~grep "a"~~
- d) ~~grep "a"~~
- e) ~~grep "a"~~
- f) ~~grep "a"~~
- g) ~~grep "a"~~

áno  
↑  
tobě  
čosi



4.

results. sh

while read LINE

do

NAME=\$(echo "\$LINE" | cut -d ";" -f 1)

MARK=\$(echo "\$LINE" | cut -d ";" -f 2)

EMAIL=\$(echo "\$LINE" | cut -d ";" -f 3)

COMM=\$(echo "\$LINE" | cut -d ";" -f 4)

TEXT=\$(echo "\$MARK" "\$COMM")

echo "\$TEXT" | mail "\$EMAIL"

done

✓ 15

5.

```
def morse_code (s):
```

```
    a = morse - 1
```

```
    d = {}
```

```
    for b in a:
```

```
        i = b.split()
```

```
        d[i[0]] = i[1]
```

```
    st = ""
```

```
    l = []
```

```
    for k in (list(s.upper())):
```

```
        l.append(d[k])
```

```
        b.append("/")
```

```
    st = "".join(l)
```

```
    return st
```

to je slyšitelné,

for e in b:  
↓ iteruje  
po známkách

↑ string

```
def morse_decode (s)
```

```
    l = []
```

```
    a = morse - 1
```

```
    d = {}
```

```
    for b in a
```

```
        i = b.split()
```

```
        d[i[1]] = i[0]
```

```
    st = ""
```

```
    for k in (s.split("/")):
```

```
        l.append(d[k])
```

```
    st = "".join(l)
```

```
    return st
```

15