

KRISTOF

$$15 + 0 + 12 + 11 + 8 = 46$$

1) echo hello > subor.txt

ln subor.txt link1.txt

cat link1.txt

ln link1.txt link2.txt

cat link2.txt

ln -s subor.txt slink1.txt

ln -s link1.txt slink2.txt

rm subor.txt

cat link1.txt

cat slink1.txt

cat slink2.txt

rm link1.txt

cat link2.txt

cat slink2.txt

mkdir dir

ln dir ~~link~~ ldir

ln -s dir sldir

ls ldir

ls sldir

→ do subor.txt vložíme řetězec "hello"

→ vytvoříme pevný link mezi subor.txt a link1.txt

→ výstup je "hello", ukazuje na obsah v subor.txt

→ vytvoříme pevný link mezi ~~subor~~^{link1}.txt a link2.txt

→ výstup je "hello", ukazuje na rovnaký inode ^{ako 1}

→ symb. link mezi subor.txt a slink1.txt

→ symb. link mezi link1.txt a slink2.txt

→ odstránim subor.txt, očakávam chybu.

→ výstup je "hello", lebo pevný link ukazuje aj na obsah inode, nie len na adresu.

→ CHYBA: vymazali sme subor.txt, teda slink1.txt neukazuje na žiaden súbor. Symbolický link

"posiela" len adresu súboru na ktorý ukazuje.

→ výstup je "hello", keďže symbolický link ukazuje na existujúci súbor s obsahom.

→ mažem link1.txt

→ výstup je "hello", medzi link1.txt a link2.txt je pevný link

→ CHYBA: podobne ako pri subor.txt, slink2.txt neukazuje na žiaden súbor. ✓

→ vyrobím rovný adresár s názvom "dir".

→ pevný link medzi dir a ldir.

CHYBA: pevný link medzi adresármi môže poroziť, pretože ~~vytvorí cyklus~~ môže vytvoriť

cykly, a tie sa vo strome vyskytovať nesmú. ✓

→ vytvorí symb. link, ktorý je poroziť, keďže len čieta cestu, takže ~~nadarmo~~ nevyraďa cykly ✓

→ CHYBA: nemám prístup k ldir, lebo neexistuje, keďže som nevyrobil žiaden pevný link

→ výstup je prázdny, keďže dir je prázdny adresár

15

2) výstup: 2
1
2
0
2
1
2

→ medzitym sa vykonávajú ešte child
procesy, ktoré ale skončia bez výstupu,
vypisujú sa len rodičovské.

↓
to mi je vysvetlenie
vysvetlenie je
to, čo ste
spravili
v príklade 1

3. a) 'baba' ✓

b) 'a' ✓

c) 'a\$' ✓

d) 'a|b' ✓

e) '[a]' ✓

f) ~~.....~~ 'a|..).*\1\$' ✓

g) '([a-z]*).*\1\$' → to splňujú všetky príklady

4) #!/bin/bash

while read line

do

meno = ~~line~~ line | cut -d ";" -f 1

mail_adresa = line | cut -d ";" -f 3

znamka = line | cut -d ";" -f 2

komentar = line | cut -d ";" -f 4

mail/##mail-adresa## <<< 'meno: \$meno znamka: \$znamka

komentar: \$komentar'

done

echo \$line &n; všade

\$(...)

↓ ↑

a toto

11

5) najprv konvertujem morse_l na slovník:

```
array_words = []
```

```
array_symbols = []
```

```
for exp in morse_l:
```

```
    x = exp.split()
```

```
    array_words.append(x[0])
```

```
    array_symbols.append(x[1])
```

to nám zlyhá na tom poslednom prvku, čo obsahuje medzeru

OK.

```
morse_dict = dict(zip(array_words, array_symbols))
```

keď už máme slovník, môžeme definovať morse-code a morse-decode

```
def morse_code(message)
```

```
    sifra = ''
```

```
    for symbol in message:
```

```
        if symbol != ' ':
```

```
            sifra += morse_dict[symbol] + '/'
```

```
        else:
```

```
            sifra += "//" # nové slovo
```

```
    return sifra
```

← message = message.upper()

→ na konci nám zostane

"/"

↓
str.join bolo treba

```
def morse_decode(message)
```

```
    message = message + ' '
```

```
    result = ''
```

```
    word = '' # jedno písmeno
```

```
    for letter in message:
```

```
        if (letter != ' '):
```

```
            i = 0
```

```
            word = word + letter
```

```
        else:
```

```
            i += 1
```

```
        if i == 2:
```

```
            result += ' '
```

kde je toto?

```
    else:
```

```
        result += list(morse_dict.keys())[list(morse_dict.values().index(word))]
```

```
        word = ''
```

```
    return result
```

rozumiem,
↓ ale je to des.
~~bez rozumiem~~

7,5