

$$10+3+6+0+4+3=26$$

1. Simona Mikovčíková

a.) Setuid a Setgit pre spustiteľné súbory

platí pre vlastníka      platí pre skupinu      x majú programy a skupiny

- prístupové práva umožnia spustiť súbor. (x - execute)

Po vykonaní exec sa UID procesu zmení na UID vlastníka súboru

Ak je vlastníkom root, súbor je spustiteľný i nastavené setuid

Po exec proces beží pod root (napr. použijeme pri zmene prihlásovacieho hesla)

b.) setgid pre adresáre

všetky nové súbory v adresáre majú pridelenú rovnakú skupinu

Novovzniknuté podadresáre majú zadaný setgid bit. abo adresári

(10)

3. ls - ukazuje nám zoznam súborov v aktuálnom adresáre

a.) ls \$X      '\$X' echo interpreтуje X ako slovo

b.) ls -f vypisala všetky súbory v adresáre (echo + ls)

\$X dereferenceuje premennej X

(6)

d.) vývodovky odšpeciálnej výško obozem \$

→ \$X = znakový súbor ⇒ \$X \* ~ echo \*

ls echo echo \*

~ výpis súborov

c.) ~~referencie na~~

\$(echo \*)

~ vykonaj prikaz ktorý je v zátvorku

echo \$X - vykoná prikaz \$X

Výplň



a) Ako  
čer?

1) . -

je ~ premennej

X reťazec 'echo \*'

b)

2) Po dereferencii

\$X dostaneme

echo echo \*

c)

3) Potom sa

rozvinie \*,

takže máme

echo echo ahoj.txt dovi.txt

4) tento prikaz sa

toto vám vykoná; číš výstup

číš

echo ahoj.txt dovi.txt

(5)

run-test script1.py

#!/bin/bash

NAZOV = \$(echo "\$1" | sed 's/\.\py\$//')

\* if test ! -d NAZOV

then

mkdir NAZOV

copytestimre

cp "\$1" test-in.txt

cat \$1 > test-out.txt

fi

(5) run-test script1.py ~~XXXXXXXXXX~~

```

#!/bin/bash
NAZOV=$(echo "$1" | sed 's/\.\py$//')
if test ! -d $NAZOV
then
    mEdit
    cp test-in.txt
    cp "$1"
    cat $1
fi

```

NAZOV → jeden alebo via argument?

mEdit → bez koncovky

cp test-in.txt → možno byť v tom predadresári?

cp "\$1" → > test-out.txt

cat \$1 → kde je spustenie?

(4)

(6) ~~aux~~

```

def txt-to-words(txt):
    x = txt.split()
    for i, item in x:
        if x[i] == ":" or x[i] == ";" or x[i] == "."
            x.remove(item)
            continue
        newx = x[i]
    
```

or text je vela riadkov/rebricov nie jeden!

(3)

anf

⑥)

def txt\_to\_words(txt):

x = txt.split()

for item in x

if x[item] == "}" or x[item] == ":"

x.remove(item)

continue

new\_x = x[item]

- (1) (10 bodov) Vysvetlite, čo znamenajú prístupové byty  
(a) **setuid** a **setgid** pre spustiteľné súbory,  
(b) **setgid** pre adresáre.

(2) (10 bodov) Na základe vašej znalosti použitých systémových funkcií *vysvetlite*, čo a ako robí funkcia **forkmap**. Na riadkoch 35–40 vidíte príklad jej použitia. Vaša odpoveď by sa nemala skladať z viet typu „na riadku *n* sa volá funkcia *foo*“ (pretože taká odpoveď preukazuje iba to, že viete čítať) ale mala by preukazovať vaše porozumenie vecí.

Je to napísané v Pythone ale to by pre Vás nemal byť problém.

Ako dlho (rádovo) si myslíte, že ten program beží?

```

1 import os      importujeme moduly
2 import sys
3 import time

4

5 def forkmap(f,zoz):
6     read_end,write_end=os.pipe()    vytvára dve s čítavým a zapisujúcim
7     read_file=os.fdopen(read_end)  do premennej read_file
8     write_file=os.fdopen(write_end,"w")  otvára čítavý koniec
9
10    názov  nov  zo zoznamu          file
11    for i,item in enumerate(zoz):  otvára
12        if os.fork()==0: nastane priečinok sme v child procese
13            write_file.write("%d %d\n" % (i,f(item)))  otvára súbor na zápis
14            write_file.flush()  vymazanie  k zápisu do súboru
15            sys.exit()  ukončenie "medzi pásmi"
16                           process
17            write_file.close()  zatváranie súboru na zápis
18
19        for x in zoz:
20            os.wait()  čaká na dokončenie child procesu
21
22    zoz_ret=[0]*len(zoz)
23
24    while True:
25        line=read_file.readline()  číta po riadkoch
26        if not line:  ak už nie je čo prečítať program sa ukončí
27            break
28        i_s,result_s=line.split()  rozdeľa v riadku na parsovanie
29        zoz_ret[int(i_s)]=int(result_s)  do zoznamu sa uložia jednotlivé znaky
30
31    return zoz_ret
32
33 def square(x):
34     # Dlhý výpočet
35     time.sleep(5)
36     return x*x
37
38 print(forkmap(square,range(100)))

```

14  
-0 zo znamy

vytvára dve s čítavým a zapisujúcim  
deskríptorm

do premennej read\_file  
otvára čítavý koniec

file

otvára súbor na zápis

sme v child procese

k zápisu do súboru

ukončenie "medzi pásmi"

process

zatváranie súboru na zápis

čaká na dokončenie child procesu

číta po riadkoch

ak už nie je čo prečítať program sa ukončí

rozdeľa v riadku na parsovanie

do zoznamu sa uložia jednotlivé znaky

1

2

funkcia zapisuje do súboru

čísla od 0 po 100 a vyskakuje

aplikuje na ne obidve mochiny

- (3) (10 bodov) Uvažujme takúto situáciu: máme spustený bash na termináli, príkaz `ls` vypisuje toto:

OK, ALE CÓ TO CÉLÉ  
AKO TO FUNBUJEZ  
ROBÍ, (3)

- (1) (10 bodov) Vysvetlite, čo znamenajú prístupové byty  
 (a) `setuid` a `setgid` pre spustiteľné súbory,  
 (b) `setgid` pre adresáre.
- (2) (10 bodov) Na základe vašej znalosti použitých systémových funkcií *vysvetlite*, čo a ako robí funkcia `forkmap`. Na riadkoch 35–40 vidíte príklad jej použitia. Vaša odpoveď by sa nemala skladať z viet typu „na riadku *n* sa volá funkcia *foo*“ (pretože taká odpoveď preukazuje iba to, že viete čítať) ale mala by preukazovať vaše porozumenie vecí.

Je to napísané v Pythone ale to by by mal byť problém.

Ako dlho (rádovo) si myslíte, že ten program beží?

```

1 import os      importujeme moduly
2 import sys
3 import time
4
5 def forkmap(f,zoz):
6     read_end,write_end=os.pipe()   ← vytvára dve s čítavým a zapisujúcim
7     read_file=os.fdopen(read_end) ← do premennej read_file
8     write_file=os.fdopen(write_end,"w") ← otvára čítavý koniec
9     write_file.write("%d %d\n" % (i,f(item))) ← vysielanie do súboru
10    write_file.flush()           ← vymazanie
11    sys.exit()                 ← ukončenie "medzi pamäť"
12    write_file.close()          ← čítavanie súboru na zapis
13    for x in zoz:
14        os.wait()               ← čaká na dokončenie child procesu
15
16    zoz_ret=[0]*len(zoz)
17
18    while True:
19        line=read_file.readline() ← read_file sa číta po riadkoch
20        if not line:           ← ak už nie je čo čítať program sa ukončí
21            break
22        i_s,result_s=line.split() ← rozdeľa v riadku na slova
23        zoz_ret[int(i_s)]=int(result_s) ← do zoznamu sa možia jednotlivé znaky
24
25    return zoz_ret
26
27
28
29
30
31
32
33 def square(x):
34     # Dlhý výpočet
35     time.sleep(5)
36     return x*x
37
38 print(forkmap(square,range(100)))

```

*po každý prvek zo zoznamu (iteruje)*

*vytvára dve s čítavým a zapisujúcim deškriptormi*

*do premennej read\_file*

*otvára čítavý koniec*

*atmosféru*

*čítavý koniec*

*otvára súbor na zapis*

*do súboru*

*ukončenie "medzi pamäť"*

*proces*

*čítavanie súboru na zapis*

*čaká na dokončenie child procesu*

*do zoznamu sa možia jednotlivé znaky*

*funkcia zapisuje do súboru*

*čísla od 0 po 100 a výpočtu*

*aplikuje na ne akom močinu*

*0\*0*

*1\*1*

*:*

*m\*m*

- (3) (10 bodov) Uvažujme takúto situáciu: máme spustený bash na termináli, príkaz `ls` vypisuje toto:

1