

1. Simona Mikovčíková

a.) Setuid a Setgit pre spustenie súboru  
↑ ↑  
platí pre všetky súby - platí pre skupinu

-prístupové práva umožnia spustiť zákon. (X-execute) a strihy

Po vykonaní exec sa UID procesu zmení na UID vlastníka suboru

Ak je vlastníkom root príbor je spustiteľný i nastavení setud

po exec proces běží pod root (napr. použijeme při zmene přihlašovacího hesla)

b.) setgid pre adresare

Všetky nové subory v adresári mají pridelenú rovnakú skupinu

Nauvěnitnute' podadresre mají zdejší setříd bit. abo adresáci

3. LS - učízuje nám zoznam súborov v aktuálnom adresári

a.)  $\hookrightarrow \$x$  '\$x\$' echo interpretuje \$x\$ ako slovo

b.) Lf-vepisala užívá soubory v adresáři ~~zpracovat~~ (echo \* & ls)  
\$ X dereferenceje premenné X

d.) úvodzony od speciálnegu všetko okrem \$

\* \$X = znakový súbor  $\Rightarrow$  \$X \* = echo \*

```
L> echo    echo *
```

$\sim$  *Vipis suborov*

c.) ~~By reference~~ na

```
$ (echo *)
```

vykonaj příkaz který je v závorce

`echo $x` - vykona prikaz `$x`

Virgals

(5)

run-test

#! bin/bash

NAZOV = \$(echo

script1.py

\$(cat NAZOV | python \$1)

\*

if test ! -d NAZOV

then

mkdir NAZOV

else

cp \$1 test-in.txt

cat \$1 > test-out.txt

fi

anf

(c)

def txt\_to\_words (txt):

x = txt.split ()

for item in x

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

x.remove ("item")

continue

newx = 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 vied typu „na riadku *n* sa volá funkcia *foo*“ (pretože taká odpoveď preukazuje iba to, že viete čítať) ale mala by preukazovať vaše porozumenie veci.

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
7     read_end,write_end=os.pipe() ← vytvára dve súkromne s čítavým a zapisujúcim
8     read_file=os.fdopen(read_end) <-- do premennej read_file
9     write_file=os.fdopen(write_end,"w") ← otvára čítavý koniec
10    write_file.write("prvok v zoznamu") ↑          ↑
11    for i,item in enumerate(zoz):                  otvára súbor na zápis
12        if os.fork()==0: nastane, ak v súčasnosti sme u child procese
13            write_file.write("%d %d\n" % (i,f(item))) ← zápis do súboru
14            write_file.flush() vymazanie
15            sys.exit() ← ukončuje "medzi pamäťou"
16                           proces
17            write_file.close() ← uzavratanie 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() ← read_file sa číta po riadkoch
26        if not line:           ab už nie je čo prečítať program sa ukončí
27            break
28        i_s,result_s=line.split() ← rozdeľuje slova v riadku na posloupnosť
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)))

```

*zoznamu*

*čítavým a zapisujúcim*  
*diskriptorom*

*čítavý koniec*

*otvára*

*zápis do súboru*

*ukončuje "medzi pamäťou"*  
*proces*

*uzavratanie súboru na zápis*

*čaká na dokončenie child procesu*

*do zoznamu sa uložia jednotlivé znaky*

*funkcia zapisuje do súboru*  
*čísla od 0 po 100 a následne*  
*aplikuje na ne akumulátor*

*0\*0*  
*1\*1*  
 $\vdots$   
*n\*n*

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

- (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 vied typu „na riadku *n* sa volá funkcia *foo*“ (pretože taká odpoved preukazuje iba to, že viete čítať) ale mala by preukazovať vaše porozumenie vecí.

Je to napísané v Pythone ale to by pre Vás nemalo 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
7     read_end,write_end=os.pipe()    vytvára ruku s čítavým a zapisujúcim
8     read_file=os.fdopen(read_end)  do premennej read_file
9     write_file=os.fdopen(write_end,"w")  otvára čítavý koniec
10    write_file.write("%d %d\n" % (i,f(item)))  náčin  nového zoznamu  file
11    write_file.flush()  vymazanie
12    sys.exit()  ukončuje "medzi pamäť"
13
14    write_file.close()  F  čítavanie súboru na zápis
15
16    for x in zoz:
17        os.wait()  čaká na dokončenie child procesu
18
19    zoz_ret=[0]*len(zoz)
20
21
22    while True:
23        line=read_file.readline()  číta po riadkoch
24        if not line:  ak už nie je čo prečítať program sa ukončí
25            break
26        i_s,result_s=line.split()  reťazec  rozobera v riadku rotparsuje
27        zoz_ret[int(i_s)]=int(result_s)  do zoznamu sa uložia jednotlivé znaky
28
29
30    return zoz_ret
31
32
33 def square(x):
34     # Dlhý výpočet
35     time.sleep(5)
36     return x*x
37
38 print(forkmap(square,range(100)))

```

*počas kódov*

*prvok zo zoznamu (iterácia)*

*vytvára ruku s čítavým a zapisujúcim deskríptorom*

*do premennej read\_file*

*otvára čítavý koniec*

*otvára súbor na zápis*

*nastane priečasne sme v child procese*

*zápis do súboru*

*ukončuje "medzi pamäť"*

*čítavanie súboru na zápis*

*čaká na dokončenie child procesu*

*ak už nie je čo prečítať program sa ukončí*

*rozobera v riadku rotparsuje*

*do zoznamu sa uložia jednotlivé znaky*

*funkcia zapisuje do súboru*

*čísla od 0 do 100 a vysporiada*

*aplikuje na ne akcie*

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

1

640

1 \* 1

3

m 7 r