

Skriptni jezici

Uvodno predavanje

Izv. prof. dr. sc. Zoran Kalafatić
Prof. dr. sc. Siniša Šegvić
Doc. dr. sc. Stjepan Groš

Zoran.Kalafatic@fer.hr, Sinisa.Segvic@fer.hr, Stjepan.Gros@fer.hr

ZEMRIS, 2017/18

Primjer 1

Potrebno je preimenovati veliku skupinu datoteka čiji oblik imena je `DSC_YYYY-MM-DD_NNNN.JPG` u novi oblik `Ljeto_YYYY_NNNN.JPG`

- rješenje 1

- ▶ otvoriti *Windows Explorer*
- ▶ klik na desnu tipku miša nad imenom datoteke
- ▶ opcija *Rename*
- ▶ urediti ime
- ▶ napraviti to za svaku od datoteka (fotografija)

- rješenje 2

- ▶ iz naredbenog retka ljuške operacijskog sustava (npr. `bash`) pozivati naredbu:

```
mv DSC_2011-07-22_1234.JPG Ljeto_2011_1234.JPG
```

● rješenje 3

- ▶ napisati *bash* skriptu koja će u petlji preimenovati sve datoteke

```
#!/bin/bash
```

```
# preimenovati DSC_YYYY-MM-DD_NNNN.JPG
```

```
# u Ljeto_YYYY_NNNN.jpg
```

```
for ime1 in DSC_2011*.JPG; do
```

```
    pom=${ime1/-??-??/}    # brisem nepotrebni dio
```

```
    ime2=${pom/DSC/Ljeto} # zamjenjujem prefiks
```

```
    mv $ime1 $ime2
```

```
done
```

Primjer 2

Potrebno je prirediti web stranicu s popisom diplomskih radova, u nekom zadanom formatu. Na raspolaganju nam je *excel* tablica s podacima: *Prezime, Ime, Naslov, Godina*

- rješenje 1

- ▶ kopirati podatke u *Word*
- ▶ uređivati polje po polje (npr. naslov kurzivom)
- ▶ spremiti datoteku u *html* obliku
- ▶ što ako se naknadno odlučimo za drugačije formatiranje?

- rješenje 2

- ▶ kopirati podatke u editor
- ▶ uređivanje formata izravno u *html* zapisu

● rješenje 3

- ▶ eksportirati podatke iz *Excel*-a u CSV formatu (*Comma-Separated Values*)
- ▶ napisati *Perl* skriptu koja će redak po redak izrezivati polja i zapisati ih u željenom *html* formatu

```
#!/usr/bin/perl -w
```

```
# diplomski u html ...
```

```
#<UL> <LI>Ime Prezime, <I>Naslov</I>, godina </LI> ... </UL>
```

```
print "<UL> \n";
```

```
while (<>){
```

```
    if(/(.*), (.*), (.*) ([0-9]{4})/){
```

```
        print "        <LI> $2 $1, <I>$3</I>, $4 </LI> \n";
```

```
    }
```

```
}
```

```
print "</UL> \n";
```

Primjer 3

Trebamo pristupati nekim *web* stranicama i prikupljati podatke. Npr. prikupiti linkove na slike koje se pojavljuju na analiziranoj stranici (``).

- jednostavno rješenje: *Python* skripta

```
# Python 3.x
import re, sys
from urllib.request import urlopen

# adresu stranice prenosimo kao argument skripte
adr = sys.argv[1].rstrip()
page = urlopen(adr).read().decode('utf8')

# nađimo slike
slike=re.findall('<img.+src="([^\"]*)">',page,re.IGNORECASE)
for s in slike:
    print(s)

print("broj slika: "+str(len(slike)))
```

Neke primjene skriptih jezika

- web aplikacije
 - ▶ PHP (Facebook), Perl (Amazon), Python, Ruby
- sistemska i mrežna administracija
 - ▶ Perl, bash, Python, Ruby
- povećanje fleksibilnosti aplikacija (ugrađeni interpreteri)
 - ▶ Lua, Python, Perl, JavaScript
- automatizacija repetitivnih poslova
 - ▶ bash, Perl, Python

Sadržaj predmeta

- ljska operacijskog sustava i pisanje skripti
- uvod u programski jezik Perl
- uvod u programski jezik Python
- *predmet nije zamišljen kao napredni tečaj, već kao upoznavanje s osnovnim konceptima koje će studenti koji budu trebali takvu vještinu lako usavršiti*

Motivacija i ciljevi – ljuska

- u svakodnevnoj praksi pojavljuje se potreba za automatiziranjem nekih poslova
 - ▶ može značajno poboljšati produktivnost
 - ▶ jednostavne skripte u *ljusci operacijskog sustava*
- ciljevi:
 - ▶ upoznati se s osnovnim naredbama
 - ▶ upoznati se s osnovnim konceptima pisanja skripti
 - ▶ znati pročitati i prilagoditi postojeće skripte

Motivacija i ciljevi – obrada teksta

- čest zadatak s kojim se susreću znanstvenici/inženjeri/programeri je obrada velikih količina podataka
 - ▶ obično u obliku teksta
 - ▶ pronalaženje zadanih uzoraka teksta, te njihova analiza ili pretvaranje u drugi oblik – prikladan za daljnju obradu
 - ▶ skriptni jezici, u kombinaciji s odgovarajućim alatima za obradu teksta, pokazuju se vrlo prikladnim izborom za ovakve zadatke
- cilj:
 - ▶ upoznati se s osnovnim konceptima i alatima za obradu teksta u skriptnim jezicima (npr. primjena *regularnih izraza*)

Motivacija i ciljevi – Perl

- u području automatizirane obrade teksta, *Perl* je vrlo raširen alat (*Practical Extraction and Report Language*)
 - ▶ u doba Interneta skriptni jezici su posebno dobili na značenju zbog podrške oblikovanju dinamičkih web stranica (generiranje *HTML*-a, mogućnost pozivanja skripti)
- ciljevi:
 - ▶ upoznati se s osnovama programskog jezika *Perl*
 - ▶ znati napisati jednostavnije skripte
 - ▶ moći pročitati (jednostavniji) tuđi kôd

Motivacija i ciljevi – Python

- *Python* je jedan od najpopularnijih programskih jezika današnjice
- sintaksa različita od C-a (C++, Java, C#, Perl)
 - ▶ vrlo jednostavna i jasna, tjera programera na urednost
 - ▶ sve češće se preporučuje kao prvi programski jezik
- izuzetno bogate biblioteke najrazličitijih alata
- vrlo široko područje primjene
 - ▶ jednostavne skripte (zamjena za ljusku OS-a ili Perl)
 - ▶ oblikovanje vrlo složenih programskih sustava (podrška OOP)
- cilj:
 - ▶ upoznati se s osnovama programskog jezika *Python*

Što je to uopće skripta?

- program vrlo visoke razine, najčešće vrlo kratak
- napisan u skriptnom jeziku (visoke razine)
- primjeri skriptnih jezika:
 - ▶ Unix ljuske (*shell*), Tcl, Perl, Python, Ruby, Scheme, Rexx, JavaScript, VisualBasic (VBScript, VBA),...
- mi ćemo upoznati:
 - ▶ Unix ljusku
 - ▶ Perl
 - ▶ Python

Osobine skriptnih jezika

- tipično se koriste za povezivanje više samostalnih programa u cjelinu (*glue language*)
- prikladni za intenzivnu obradu (manipulaciju) teksta
- omogućuju manipulaciju datotečnim sustavom
- često se koriste za pisanje programa usmjerenih specifičnim potrebama korisnika
- obično se ne radi o složenim programima – relativno jednostavan razvoj
- podrška oblikovanju prilagođenog korisničkog sučelja
- portabilnost (Unix, Windows, Mac)
- program se *interpretira*

Prednosti skriptnih jezika

prednosti skriptnih jezika u odnosu na *uobičajene* jezike (Java, C/C++)
– *sustavski* programski jezici

- kraći programi
 - ▶ brži razvoj programa
 - ▶ manja mogućnost pogreške
- brži ciklus provjere:
 - ▶ uređivanje koda i testiranje (bez prevođenja i povezivanja)
- nema deklaracija varijabli
 - ▶ ali zato puno provjera tijekom izvođenja programa!
- alati za povezivanje programa ("lijepljenje")
- alati za obradu teksta
- jednostavna izgradnja korisničkog sučelja (GUI)
- alati za Web programiranje

- svaka kategorija jezika ima svoje područje primjene
- *skriptni jezici*
 - ciljaju na čim bržu i jednostavniju izgradnju programa
 - ▶ koriste se bogate biblioteke gotovih *komponenti* koje se povezuju u cjelinu
 - ▶ *komponentno* programiranje
- *sustavski jezici*
 - nastoje osigurati čim veću učinkovitost konačnog kôda
 - ▶ mogućnost upravljanja svim detaljima izvedbe
 - ▶ prikladni za izradu komponenti koje će se povezati u skriptnom jeziku, pogotovo za računski zahtjevne dijelove funkcionalnosti

Primjer povezivanja komponenti

- napisati program koji učitava niz slika i prati objekte u tom slijedu slika
- program za analizu slika napisat ćemo u C-u, a slike se mogu čitati sa standardnog ulaza (`stdin`) – na koji se može preusmjeriti sadržaj niza slikovnih datoteka ili pak slike dobivene s digitalizatora slike
- rezultat (slika s označenim objektom) može se zapisati na standardni izlaz (`stdout`) koji se može preusmjeriti u datoteku ili u program za prikaz slike
- kratka skripta koja pokreće te komponente:

```
cat slika*.ras | prati_objekt | display  
poziva se s:  
./pokreni_pracenje
```

- ako napišemo i program (u C-u) koji pristupa digitalizatoru slike, preuzima sliku po sliku, te ih u odgovarajućem formatu prosljeđuje na `stdout`, njegov izlaz možemo ulančati s ulazom programa za praćenje objekata:

```
grab | prati_objekt | display
```

- možemo napisati program (u C-u) za predobradu ulazne slike (npr. glađenje Gaussovim filtrom zadanog parametra σ) koja sa `stdin` čita sliku, a glađenu sliku zapisuje na `stdout`:

```
grab | gladixy -s 1.5 | prati_objekt | display
```

- ako se pojavi potreba za demonstracijom programa, lako je pripremiti skriptu koja učitava nekoliko karakterističnih sekvenci iz datoteka, i pokreće program za praćenje s odgovarajućim parametrima, te ispisuje na ekranu odgovarajuće poruke
- možemo dodati i grafičko korisničko sučelje (npr. složeno u Pythonu)

Klasifikacija programskih jezika

- za razredbu programskih jezika mogu se primijeniti različiti kriteriji
- jezici s dinamičkim odnosno statičkim dodjeljivanjem tipova (*dynamically vs statically typed languages*)

- Python (dinamički):

```
c = 1           # c je integer
c = [1, 2, 3]   # c je lista
```

- C (statički):

```
double c; c = 5.2;    /* c prima samo double */
c = "a string...";    /* compiler error */
```

- slabo odnosno jako (strogo) tipizirani jezici
(*weakly vs strongly typed languages*)

- Perl (slabo):

```
$b = '1.2'  
$c = 5*$b;    # implicitna pretvorba tipova:  
              # '1.2' -> 1.2
```

- Python (jako):

```
b = '1.2'  
c = 5*b      # nema implicitne pretvorbe tipova  
              # rezultat je ponavljanje niza  
              # '1.21.21.21.21.2'
```

- interpretirani odnosno prevedeni jezici
- jezici visoke odnosno niske razine (Python–C)
- jezici vrlo visoke razine nasuprot jezika visoke razine (Python–C++)
- skriptni nasuprot sustavskih jezika (scripting vs system languages)

Organizacija nastave

- predavanja – 2 sata tjedno
- laboratorijske vježbe – 3 bloka = 3 termina
 - ▶ *samostalan* rad – rješavanje zadataka = “domaće zadaće”
 - ★ predaja **tuđih rješenja** je povreda *Kodeksa ponašanja studenata FER-a*
 - ▶ rješenja se predaju preko *Ferka* – preduvjet za pristupanje laboratorijskoj vježbi
 - ▶ na vježbama se piše “ulazni test” – zadaci s ponuđenim odgovorima
 - ▶ rješenja se demonstriraju asistentu – ocjenjuje se
 - ▶ tijekom laboratorijskog termina piše se test na računalu
 - ★ dobiva se novi zadatak koji treba isprogramirati na licu mjesta i predati kroz *Ferko*
- provjere znanja
 - ▶ međuispit
 - ▶ završni ispit
 - ▶ laboratorijske vježbe

Kalendar nastave

OŽUJAK					
	1	2	3	4	
P	5	12	19	26	
U	6	13	20	27	
S	7	14	21	28	
Č	1	8	15	22	29
P	2	9	16	23	30
S	3	10	17	24	31
N	4	11	18	25	1.4.

TRAVANJ					
	L1	6	7	MI	MI
	2	9	16	23	30
	3	10	17	24	
	4	11	18	25	
	5	12	19	26	
	6	13	20	27	
	7	14	21	28	
	8	15	22	29	

SVIBANJ					
	MI	8	L2	10	11
P		7	14	21	28
U	1	8	15	22	29
S	2	9	16	23	30
Č	3	10	17	24	31
P	4	11	18	25	
S	5	12	19	26	
N	6	13	20	27	

LIPANJ					
	12	L3	ZI	ZI	
	4	11	18	25	
	5	12	19	26	
	6	13	20	27	
	7	14	21	28	
	1	8	15	22	29
	2	9	16	23	30
	3	10	17	24	

Kontinuirano praćenje nastave

- laboratorijske vježbe:
 - ▶ 30 bodova
- međuispit:
 - ▶ 30 bodova
- završni ispit
 - ▶ 40 bodova
- **uvjet** za izlazak na završni ispit
 - ▶ 50% bodova na laboratorijskim vježbama (15 bodova)
- uvjet za polaganje predmeta:
 - ▶ ukupno ostvarenih barem 50 bodova
 - ▶ **barem 30% bodova** ostvarenih na završnom ispitu (12 bodova)

Ispitni rokovi

- **uvjet** za izlazak na ispit
 - ▶ 50% bodova na laboratorijskim vježbama (15 bodova)
- bodovi na ispitnom roku
 - ▶ laboratorijske vježbe: 30 bodova
 - ▶ ispit: 70 bodova
- uvjet za polaganje predmeta:
 - ▶ ukupno ostvarenih barem 50 bodova
 - ▶ **barem 50% bodova** ostvarenih na pismenom ispitu (35 bodova)

Ocjene – u oba načina polaganja ispita

Ocjena	Minimalan broj bodova
Dovoljan (2)	50
Dobar (3)	62
Vrlo dobar (4)	75
Izvrstan (5)	88

Literatura I

- preporučena literatura:

- ▶ Stephen Kochan, Patrick Wood,
Unix Shell Programming, 3rd edition, Sams, 2003
- ▶ Randal L. Schwartz, Tom Phoenix, brian d foy,
Learning Perl, 5th edition, O'Reilly, 2008
- ▶ Mark Lutz,
Learning Python, 4th edition O'Reilly, 2009
- ▶ Zoran Kalafatić, Antonio Pošćić, Siniša Šegvić, Julijan Šribar,
Python za znatiželjne, Element, 2016
- ▶ Zoran Kalafatić,
Skriptni jezici – Materijali za predavanja, FER, 2012

- slajdovi će biti raspoloživi na stranici predmeta

<https://www.fer.hr/predmet/skrjez>

Literatura II

- mnoštvo literature na Internetu

- ▶ Mendel Cooper, *Advanced Bash-Scripting Guide*
<http://tldp.org/LDP/abs/html/>
- ▶ James Lee, Simon Cozens, Peter Wainwright, *Beginning Perl*
<http://www.perl.org/books/beginning-perl/>
- ▶ <http://www.perl.org>
- ▶ <http://www.python.org>