LUA DASTURLASH TEXNOLOGIYAASIDA GLOBAL O'ZGARUVCHILAR HAMDA MUSTAQIL TARJIMON TILI HAQIDA UMUMIY TUSHUNCHALAR (LUA DASTURLASH TILI BO'YICHA)

Namangan davlat universiteti Amaliy matematika va raqamli texnologiyalar kafedrasi stajor o'qituvchisi
Zokirova Nargiza Sadriddin qizi
Namangan davlat universiteti Amaliy matematika va raqamli texnologiyalar kafedrasi stajor o'qituvchisi
Muradova Kamola Akramjanovna

Anotatsiya: Ushbu maqolada Lua dasturlash texnologiyalarida global o'zgaruivchilar va bu o'zgaruvchilarning dasturlkardagi ro'llaeri haqida so'z brogan bundan tashqari, Lua dasturlash texnologiyasi mustaqil tarjimon tili sifatida qanday ishlart amalga oshirilgani haqida umumiy ma'lumotlarf va tushunchalar berib o'tilgan. Lua dasturlash tili UNIX tizimida ishlay olishi va uning maqsadlarini amalga oshira olishi haqida umumiy tushunchalar berib o'tilgan

Kalit so'zi: Lua dasturlash tilidagi global o'zgaruvchilar, global o'garuvchi nime ekanligi, mustaqil tillar va mustaqil tarjimon haqidaz tushuncha

ПРЕПОДАВАТЕЛЬ-СТАЖЕР КАФЕДРЫ ПРИКЛАДНОЙ МАТЕМАТИКИ И ЦИФРОВЫХ ТЕХНОЛОГИЙ НАМАНГАНСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Закирова Наргиза Садриддин кизи .
Докторант кафедры прикладной математики и цифровых технологий Наманганского государственного университета
Мурадова Камола Акрамджановна.

Аннотация: Помимо разговора о глобальных переменных в технологиях программирования Lua и их роли в программировании, в этой статье также дается общая информация и понимание того, как технология программирования Lua работает как независимый язык-интерпретатор. Дано общее представление о том, как язык программирования Lua может работать в UNIX и как он может достигать своих целей.

Ключевые слова: глобальные переменные в языке программирования Lua, что такое глобальная переменная, независимые языки и понимание независимых интерпретаторов.

INTERN TEACHER OF THE DEPARTMENT OF APPLIED MATHEMATICS AND DIGITAL TECHNOLOGIES OF NAMANGAN STATE UNIVERSITY

Zakirova Nargiza Sadriddin qizi
Doctoral student of the Department of Applied Mathematics and Digital Technologies of Namangan State University
Myradova Kamola Akramjanovna

Abstract: In addition to talking about global variables in Lua programming technologies and their roles in programming, this article also provides general information and understanding of how Lua programming technology works as an independent interpreter language. . A general understanding of how the Lua programming language can work on UNIX and how it can achieve its goals is given

Keywords: Global variables in the Lua programming language, what is a global variable, independent languages, and an understanding of independent interpreters

Global o’zgaruvchilar e’lon va u haqid ma’imo berilishiga muhtoj emas; dasturlarda siz uladan foydalanishingiz mumkin. Bizga ma’lum bo’lgan holda bu ishga tushirilmagan o’zgaruvchiha kirish xato emas, natijada siz faqat nol (nill)ga erishasiz:.

```
print(b) --> nil
b = 10
print(b) --> 10
```

Agar siz global o’zgaruvchiga nol(nill) qo’yiladigan bo’lsa, u holda Lua dasturlash tilida o’zgaruvchi hech qachon ishlatilmagandek, yoki umuman mavjud emasdek ma’lumotga ega bo’linishi mumkin:

```
b = nil
print(b) --> nil
```

Ushbu berilgan topshiriqdan so’ng, Lua dasturlash tili oxir oqibat ushbu o’zgaruvchi egallagan xotiranin qaytarib olish mumkin bo’ladi.

Mustaqil tarjimon (shuningdek, umumiy manba sifatida fayl nomidan keyin lua.s yoki oddiygina bajariladigan fayl omidan keyingi o’rinda layning berilgan nomidan keyin lua deb atab ketishni o’zi kifoya qiladi) Lua dasturlash tilidan bevosita foydalanish imkonini berivchi kichik dastur sifatida qarashimiz mumkin bo’ladi.

Berilgan ma’lumotlaga ko’ra, bizga berilgan tarjimon fayllarni yuklanganda, hamda berilgan fayllar # belgisi bilan boshlangan bo’lsa, shu belgilangan birinchi qarotni o’tkazib yuboradi. Bu Lua dastgurlash tilidan UNIX tizimlarida script tarjimoni

sifatida foydalalnish imkonini beradi. Agar siz skriptingizni shunga o'xshash narsalar bilan boshlasangiz mumkin bo'ladi.

```
#!/usr/local/bin/lua
```

(tarjimon /usr/local/bin ichida bo'lsa) yoki

```
#!/usr/bin/env lua,
```

Keyin Lua dasturlash tili tarjimonini aniq ishga tushirilmasdan skriptingizni bevosita ishga tushirishingiz mumkin bo'ladi. Tarjimon quydagicha holatda chiqariladi:

```
lua [options] [script [args]]
```

Barcha parametrlar ixtiyoriy holatda amalga oshirishimiz mumkin. Yuqorida aytib o'tganimizdek, lua dasturlash tilini argumentlarsiz ishga tushurganimizda, u interaktiv rejimga o'tishi mumkin bo'ladi.

-e ko'rinishdagi holatlar quydagi ko'rsatilgandek, to'g'ridan – to'g'ri buyruq satrida kodni ko'rsatishga imkon beradi.

```
% lua -e "print (math.sin(12))" --> -0.53657291800043
```

(UNIX qobiq qavslarni tahlil qilinishiga yo'l qo'ymaslik uchun qo'shtirnoqlar talab qilinadi.) -1 operatsiya kutubxonbasi yakunlashimiz mumkin bo'ladi. Yuqorida ko'rsatib o'tilganidek, -I qolgan argumentlarni qayta ishlagandan so'ng tarjimonni interaktiv rejmga ya'ni shaklga o'tkazishi mumkin bo'ladi. Shundan keyin esa keying habarlarni beruvchi lib ni yuklay oladi, keyin esa x=10 ni taniy oladi va nihoyat biz amal qilishimiz mumkin bo'lgan interaktiv rehimga o'tadi.

```
% lua -i -llib -e "x = 10"
```

Interfaol rejimdan olib qaraydigan bo'lsak siz tenglik belgisi bilan boshlanadigan qatorlarni so'ng ifodani qiymatini chop etishingiz mumkin:

```
> = math.sin(3) --> 0.14112000805987
> a = 30
> = a --> 30
```

Bu hususiyatlardan lekib chiqadigan bo'lsak Lua dasturlash tilidan kalkulyator sifatida ham foydalanish imkoniyatini beradi. Argumentlarni bajarishdan oldin tarjimon LUA_INIT_5_2 deb nomlangan muhit sifatida o'zgaruvchilarni qidirib topadi yoki bunday o'zgaruvchilar bo'lmasa LUA_INIT.deb nomlangan tarjimonga murojatqiladi. Agar ushbu o'zgaruvchilardan biri mavjud bo'lsa va uning qiymatlari @filename bo'lsa, tarjimon esa bu fayllarni ishga tushiradi. Agar LUA_INIT_5_2 (yoki shu bilan bir qatorda LUA_INIT) aniqlangan bo'lsada, lekin "@" kodini o'z ichiga oladi deb hisoblaydi va uni bajaradi. LUA-INIT tarjimonini sozlash uchun juda

kata imkoniyatlarni taqdim etiladi, chunki konfiguratsiya paytida Lua dasturlash tilining to'liq quvvat bilan sihalshini o'z ichiga oladi. Biz bu bilan bir qatorda paketlarni yuklashimiz, joriy yo'llarni o'zgartirishimiz, o'z funksiyalarini belgilashimiz, funksiyalarni nomini o'zgartirish yoki yo'q qilish malumatlarini olishi mumkin va h.z. Skript o'z argumentlarini global o'zgaruvchilar *arg* sifatida qabul qilishimiz mumkin bo'ladi. Agar bizda ma'lumot sifatidan berilgan *%lua script abs* sifadi ma'lum qilingan bo'lsa, tarjimon skriptni amalini bajarishdan avval barcha boshqa berilgan buyuruqlar qatori argumentlari bilan *arg* jadvalini yaratib olish maqsadga muvofiq bo'ladi. Skript nomi 0 indeksioda, bir qator argument (va misol uchun *"a"*) berilgan birinchi indeksda joylashgan holatda ko'rsatiladi hamda qolgan skriptlar shu qaroti amalga oshirib boriladi. Biz ko'rib chiqqan holatlarda kelib chiqqan holda manfiy indekslarini ham joylashtirishimiz mumkin, chunki berilgan holda script nomidan oldin paydo bo'ladi. Misol uchun quydagi ma'lumotni ko'rib chiqishimiz mumkin:

```
% lua -e "sin=math.sin" script a b
```

Tarjimon argumentlarni quydagicha ko'rinishda amalga oshirish mumkin:

```
arg[-3] = "lua"
arg[-2] = "-e"
arg[-1] = "sin=math.sin"
arg[0] = "script"
arg[1] = "a"
arg[2] = "b"
```

Ko'pincha script faqat ijobiy indekslardan foydalaniladi (misol uchun *arg[1]* va *arg[2]*). Lua 5.1 dan boshlab, stript o'z argumentlarini … (uch nuqta) ifodasi yordamida ham qabuil qilishi mumkin. Skriptning barcha argumentlari o'z joyida beriladi (biz shunda o'xshash iboralarni 5.2-bo'limda muhokama qilamiz)

O'tilgan mavzularga doir mashqlar

1.1-mashq

Faktarial masalani ishga tushuruvchi dastuirni yarating. Agar manfiy son kiritilgan taqdirda dasturingizda qanday o'zgarish yoki qanday hatoliklar bo'ladi? Shu xatolikni oldini olish uchun masalani qay darajada o'zgartirilishi kerak bo'ladi.

1.2-mashq

Keying masalada twice buyrug'ini ishga tushuring. Faylni birichi marotaba ishga tusahirganingizda -1 va yana bir marotaba ishga tushirilganida esa dofile bilan yuklang. Qaysi bir amal tezroq va aniqroq natijaga erishilishi aniqlansin.

Foydalanilgan adabiyotlar:

Adabiyotlar:

1. “Lua dasturlash texnologiyasi” qo’llanma N.Zokirova G.Yunusova
2. Dasturlash texnologiyalari Nazarov Sh G.Ivanova, S.Gaynazarov

Internet manbalar:

3. https://www.youtube.com/watch?v=5UCg7ca8ogE
4. https://www.youtube.com/watch?v=SBL0POpUYvQ