

Operativni sistemi - Uvod

VELJKO PETROVIĆ (PVELJKO@UNS.AC.RS)

Administrativni detalji

O predavaču

- Veljko Petrović
- pveljko@uns.ac.rs
- NTP 330
- Dok traje COVID-19 pandemija ne očekujte da me tu nađete
- Preliminarni termin konsultacija: četvrtak u 16:30 ili po dogovoru.
- Uvek je dobra ideja da se najavite za konsultacije.
- Ne zaboravite: uvek možete da mi pišete na Teams-u.

Asistenti

- Aleksić Katarina
- Andrejević Aleksandar
- Gojić Gorana
- Radović Luka
- Simić Svetislav
- Turović Radovan

Materijali

- Operativni Sistemi, Problemi i Struktura prof. Hajdukovića
- Sav materijal je dostupan na Canvas-u
- Glavni izvor za predmet je, predvidivo udžbenik.
- Korisni su i slajdovi koje gledate. I oni će biti na Canvas sajtu.

O predavanjima

- Evidencija prisustva postoji i automatska je.
- Nemojte da se samo logujete a onda ignorišete predavanje.
- Ja ću periodično nekome odabranom potpuno nasumično postaviti pitanje.
- Očekujem odgovor u roku od minut-dva. Ako ga ne dobijem interpretiram to kao da niste bili tu.
- Ovo služi da one koji su tu i obraćaju pažnju adekvatno nagradim bodovima: na raspolaganju mi je 10 bodova što nije za bacanje.
- Nikada nije pogrešno vreme da se postavi pitanje ili napravi komentar. Dižite ruku, pišite u chat, pišite mi na DM... Kada držim predavanja uživo koristim vaše izraze lica da ukapiram kada sam išao previše brzo ili sporo. Sada to ne mogu tako, tako da vas molim da mi pomognete da vam pomognem.

O vežbama

- Automatska evidencija pristupa
- Samostalna izrada zadataka
- Apsolutno se očekuje da se *spremite za izradu zadataka* pre svake vežbe
- Počinju od iduće nedelje.
- *Morate* imati spremno okruženje za rad.
- Asistenti imaju instrukcije da vam traže (po nasumičnom principu) da podelite sliku ekrana sa vremena na vreme. Radite to pod obavezno. Ako to ne uradite blagovremeno, pisaćće vam se ekvivalent odsustva sa vežbi. Plus, ako radite na vežbama i to se piše i možete osvojiti do 10 bonus bodova *i tu*.
- Više o tome šta je 'okruženje za rad' kasnije danas.

Formiranje ocene

Broj bodova	Ocena
51-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Odakle bodovi?

- Predispitne obaveze (do 70 bodova)
- Ispitne obaveze (do 30 bodova)
- Bonus bodovi
 - Do 10 sa vežbi (bodovi primenjivi samo na onih 70 bodova predispitnih obaveza)
 - Do 10 sa predavanja (bodovi primenjivi samo na onih 30 bodova parcijalnih ispita)
 - Bonus bodovi *ne mogu* pomoći da se premaši maksimum.

Bodovi predispitnih obaveza

Obaveza	Opis	Bodovi
Test T1234	Konkurentno programiranje	do 40
Složeni oblik vežbi SOV	Konkurentno programiranje (konkurentni problem)	do 30

Tehnika izrade predispitnih obaveza

- Dobićete vrlo ograničeno vreme i zadatak online.
- Uradite i predate zadatak **bez prepisivanja**.
- Kada to uradite *ne dobijate bodove*.
- Umesto toga, bodove dobijate u terminima *odgovaranja i odbrane*.
- Ono što se ocenjuje je šta možete da uradite, objasnite, i odgovorite na odbrani, *ne* zadatak.
- Čemu onda zadatak? Daje nam nešto o čemu da pričamo: jedino postoji zato što bi inače morali da odgovarate cca 4 sata a za to nema vremena. Ovako doneste uzorak rada i budete ocenjeni na vašoj sposobnosti da kažete kako bi ga modifikovali, kako bi ga podesili, i šta bi sa njim radili.

Termini odbrane i odgovaranja

- T1234 (ne pitajte za ime): 40 bodova. 08.05., 09.05.
- SOV: 30 bodova. 12.06. 13.06
- Kada se radi SOV možete da radite zadatak težine T1234 umesto (težeg) zadatka za SOV. Ako to radite stari bodovi, šta god da ste dobili, se brišu i imate ono što osvojite na odbrani novog zadatka.

Kako položiti?

- Prolaznu ocenu možete imati samo ako važi svaki od sledećih uslova:
 - $T1234 + SOV \geq 36$
 - $Ispit \geq 16$
 - $T1234 + SOV + Ispit \geq 51$

Kako pasti?

- Ako imate manje od 36 bodova sa predispitnih obaveza onda su svi bodovi koje imate nevažeći i morate predmet slušati opet iduće godine.
- Nemojte dozvoliti da ovo budete vi, molim vas.

Ispit

- Održava se u ispitnom roku
- Nosi najviše 30 bodova
- Namenjen je *isključivo* studentima koji na predispitnim obavezama imaju barem 36 bodova
- Integralni ispit obuhvata celo gradivo.
- *Mora se prijaviti ispit.*
- Radi se na papiru, ima četiri pitanja i traje 60 minuta.

Struktura ispita

1. Prvo pitanje nosi **6 bodova**, odnosi se na celo gradivo i biće bazirano na principu *zaokruživanja*. Ovo pitanje će uvek biti bazirano na listi pitanja koja ću objaviti na Canvas-u.
2. Drugo pitanje nosi **10 bodova**, odgovara se u okviru jednog do dva pasusa i mora biti sa liste pitanja koja ću objaviti.
3. Treće i četvrto pitanje nose zajedno **14 bodova** i to tipično, ali ne garantovano, po **7 bodova** i potpuno su slobodne forme: odnose se na celokupno gradivo predmeta bilo sa predavanja, udžbenika, slajdova i zahtevaju da se mogu povezati različite činjenice naučene u okviru predmeta i da se o njima može rezonovati.

Šta neće nikada biti na ispitu

- Na ispitu nikada neće biti bilo koje pitanje koje očekuje da se kod uči napamet.
- Ako se, nekim slučajem, u pitanju pomene klasa ili neki algoritam, onda se isključivo misli na namenu te klase odnosno algoritma, na to kako radi i zašto postoji ali nikada i nikako koje su metode, šta tačno nasleđuje ili bilo šta slično.
- Ovo garancija se odnosi na sva četiri pitanja i na sve rokove, zauvek.

Primer kompletnog ispita sa odgovorima

- 1. Koja su validna stanja binarnog semafora?

- a) 0 b) 1 c) 2 d) SLOBODAN

- e) ZAUZET f)-1 g) 3

- 2. Šta je mrtva petlja?

- Mrtva petlja nastaje kada više niti/procesa pristupaju deljenim resursima koji su međusobno zavisni i kojima se pristupa u režimu isključivosti. Ono što ovi uslovi omogućavaju jeste da se procesi/niti dovedu u situaciju gde za svaku nit/proces važi da čeka neku drugu nit/proces i grafik ovih zavisnosti formira zatvoren ciklus. To znači da niti/procesi nisu aktivne no, su u istanju čekanja i iz tog stanja nikako ne mogu izaći: odatle naziv 'mrtva petlja.'

Primer kompletnog ispita sa odgovorima

- 3. Kakav format diska bi izabrali za particiju koja čuva video snimke sistema za video-nadzor pod uslovom da želite maksimum iskorišćenja prostora. Obrazložiti vaš odgovor.
■ Koristio bih nekakav fajl sistem koji koristi kontinualne datoteke. Ovo je dobra ideja zato što onda nikako ne bih imao blokove, pokazivače, ili bilo šta slično. Samo metapodatke u deskriptoru datoteke i dužinu datoteke praćenu sadržajem datoteke, redom. Ne bih morao da se brinem o produženju datoteke pošto se ovo ne bi dešavalo. Eksterna fragmentacija bi, takođe bila relativno mali problem, budući da bih imao fajlove predvidive dužine. Kada bih birao veličinu bloka za ovaj disk, birao bih veliki blok, budući da ne očekujem male datoteke uopšte.
- 4. Ako bi imali pristup direktno disku (blokovima) koliko bi vam minimalno trebalo pristupa disku da promenite vlasništvo nekom fajlu? Obrazložite vaš odgovor.
■ Odgovor zavisi od fajl sistema, i okolnosti. Ako, kao na predavanjima, koristimo ext2fs, onda treba sigurno da pročitamo superblok (1 čitanje), a onda iz njega da dobijemo tabelu inoda (1 čitanje), pa onda u njoj da nađemo sadržaj root direktorijuma (1 čitanje), pa u root direktorijumu nađemo broj inode nekog, bilo kog fajla, pa onda nađemo sadržaj te inode (1 čitanje), a u toj inodi, odnosno deskriptoru fajla se čuva tabela pristupa. Tako da je odgovor oko, minimalno, 4 čitanja i jedno pisanje.
- **(Napomena:** Molim vas, vodite računa da je ovaj odgovor malo neprecizan i preskače barem jedno čitanje (za tačan odgovor, vidite pretavanje 08.4) ali da je ključna stvar za ovo specifično pitanje da je proces razmišljanja i rezonovanja korektan.)

Prepisivanje

- Ako se utvrdi da je neko prepisivao ili koristio bilo kakva nedozvoljena sredstva na proveri znanja bilo koje vrste:
 - Dobija se 0
 - Gubi se mogućnost bilo kakvog popravljavanja rezultata.
 - U zavisnosti od okolnosti može se početi disciplinski postupak na fakultetu.

Molim vas, molim vas *nemojte*. Oko svega možemo da se dogovorimo, sve može da se adaptira vašim potrebama naročito u ovim teškim vremenima, ali kada je prepisivanje u pitanju niko od vaših nastavnika na ovom predmetu nema smisao za humor.

Potpis

- Svaki student koji je dolazio na nastavu makar minimalno uredno dobija potpis.

Tehničko okruženje

KAKO BITI SPREMAN ZA VEŽBE I PRAĆENJE PRIMERA SA PREDAVANJA

Tehničko okruženje

- Na ovom kursu se koristi Linux.
- Iako se primeri *mogu* kompajlirati na Windows-u ovo *nije* preporučeno.
- Zašto? Naleteće te na suptilne nekompatibilnosti koje su takve da niste dovoljno dobri programeri da ih otklonite.
- Ovo nije prva generacija: *svako* ko je koristio Windows u ovom kursu je naleteo na problem.
- Od softvera se koristi GCC (kao kompajler), i opciono, CodeBlocks kao integrisano okruženje.
- Možete da instalirate Linux na vaš računar kao jedan od operativnih sistema i koristite to. Ali, ako vam je primaran OS Windows i ne želite da instalirate nešto novo, može i tako. Onda morate raditi u virtuelnoj mašini.

Virtuelna mašina

- Trebaće vam VirtualBox <https://www.virtualbox.org/> koji se instalira kao i svaka druga aplikacija. Dostupan je za svaki operativni sistem.
- Dalje, trebaće vam slika virtuelne mašine. Ako želite verziju koja je ista onoj u FTN laboratorijama, možete je naći na: <http://www.acs.uns.ac.rs/sr/filebrowser/download/3984950>
- Na njoj neće raditi neki primeri naprednog C++-a zato što ima staru verziju kompajlera. Vaši zadaci će biti tako napravljeni da mogu da se reše na njoj ali rizikujete probleme na samim vežbama i gubljenje bonus bodova. Ako želite noviju verziju (preporučeno) skinite ISO za neki moderniji Linux.

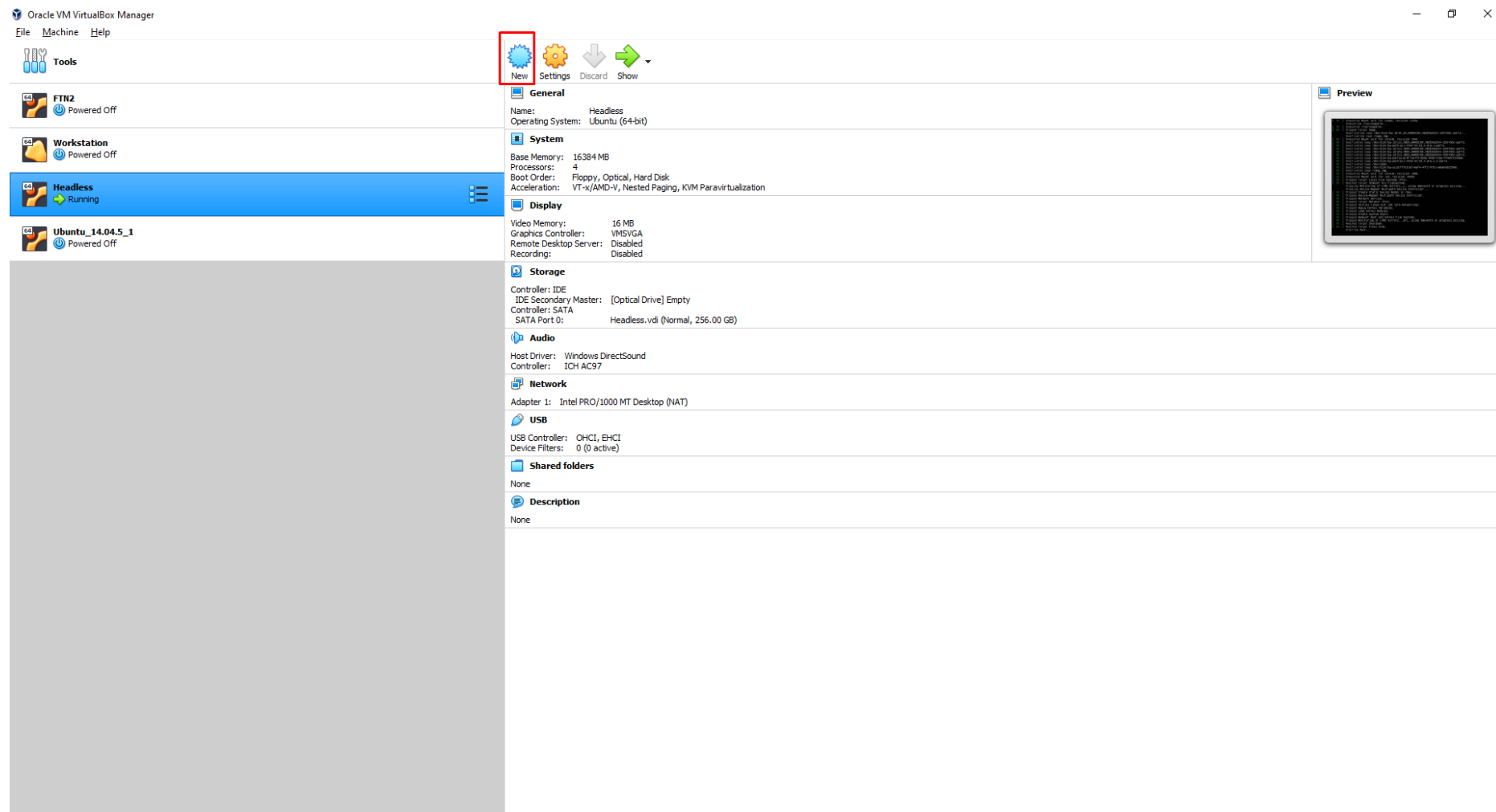
Neki moderniji Linux

- Ako imate omiljenu distribuciju: slobodno. Možete ispratiti sve u ovom kursu ako koristite ništa više nego editor i GCC kompajler.
- Ako nemate: Ubuntu Desktop najsvežija verzija: <https://ubuntu.com/#download> je gotovo sigurno ono što želite. Ako imate problem sa RS mirror-om (ja imam) <https://launchpad.net/ubuntu/+cdmirrors> je link koji može pomoći. Alternativa je download preko BitTorrent klijenta koji je obično najbrži.
- Kada preuzmete ISO napravite virtuelnu mašinu i instalirate taj ISO kao na bilo koji računar.

Preduslovi za instalaciju

- Trebaće vam instaliran VirtualBox. Skida se i instalira kao bilo koja druga aplikacija.
- Zatim morate preuzeti ISO sliku. Mi ovde koristimo svež Ubuntu 20.10. Fajl koji sam ja preuzeo se zove: ubuntu-20.10-desktop-amd64.iso i ima SHA256 sumu od 3ef833828009fb69d5c584f3701d6946f89fa304757b7947e792f9491caa270e.
- Zbog toga što mirror u Srbiji ima problem sa sertifikatima trenutno, možda budete imali problem sa preuzimanjem. To se da rešiti tako što odete na link sa mirror-ima i odaberete neki iz neke obližnje zemlje. Ako imate to podešeno, najbrže se skida preko BitTorrent protokola.
- Kada imate oba fajla, imate sve što vam treba da počnete instalaciju.

Korak #1 Virtuelna mašina



Korak #1 Virtuelna mašina

?

×

← Create Virtual Machine

Name and operating system

Name:

Machine Folder:

Type:

Version:

Memory size

8192 MB

4 MB 32768 MB

Hard disk

☐ Do not add a virtual hard disk

☒ Create a virtual hard disk now

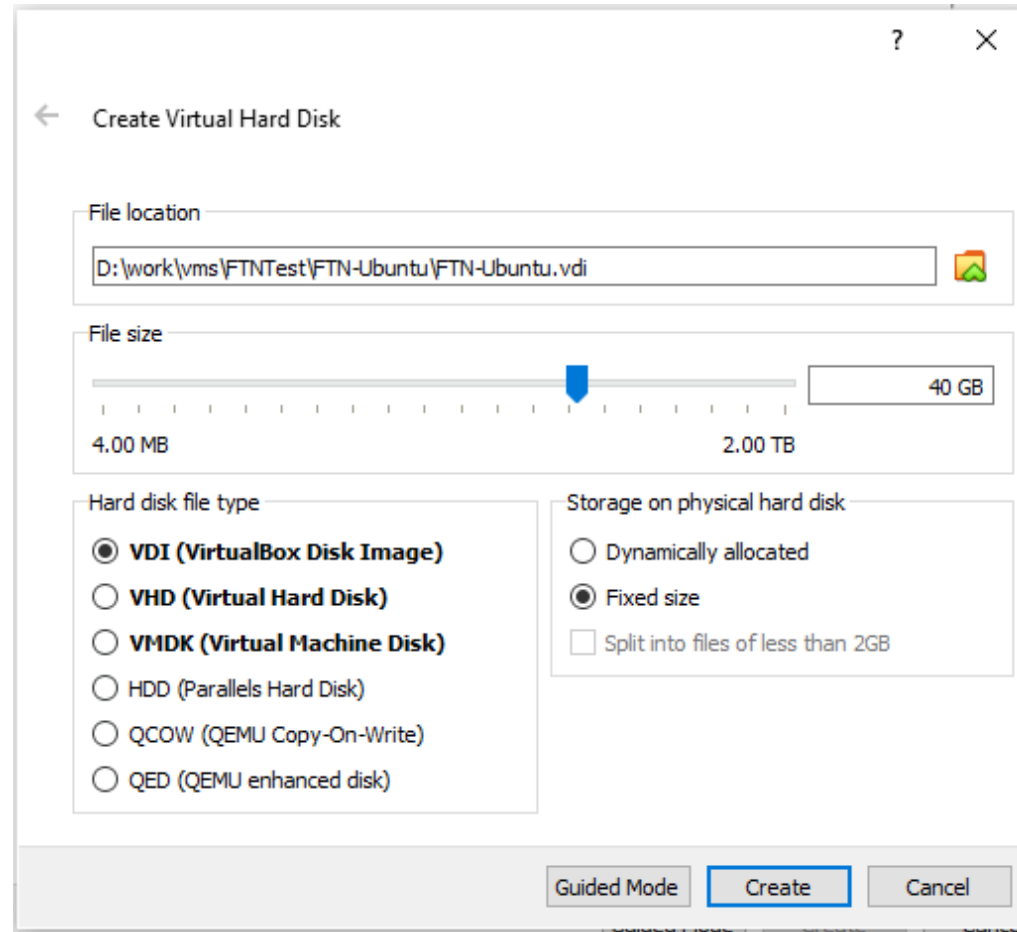
☐ Use an existing virtual hard disk file

Guided Mode **Create** Cancel

Korak #1 Virtuelna mašina

- Ime je proizvoljno
- Lokacija je takođe proizvoljna ali vodite računa da će OS pisati dosta po lokaciji mašine tj. njenom virtuelnom disku tako da morate doneti odluku da li da ga smestite na SSD ako ga naročito čuvate.
- Količina memorije zavisi od vaše mašine i načina na koji koristite vašu virtuelnu mašinu. Generalno govoreći stavite što više. 8GB (koliko je ovde stavljeno) je generalno dosta, ali više je uvek bolje.

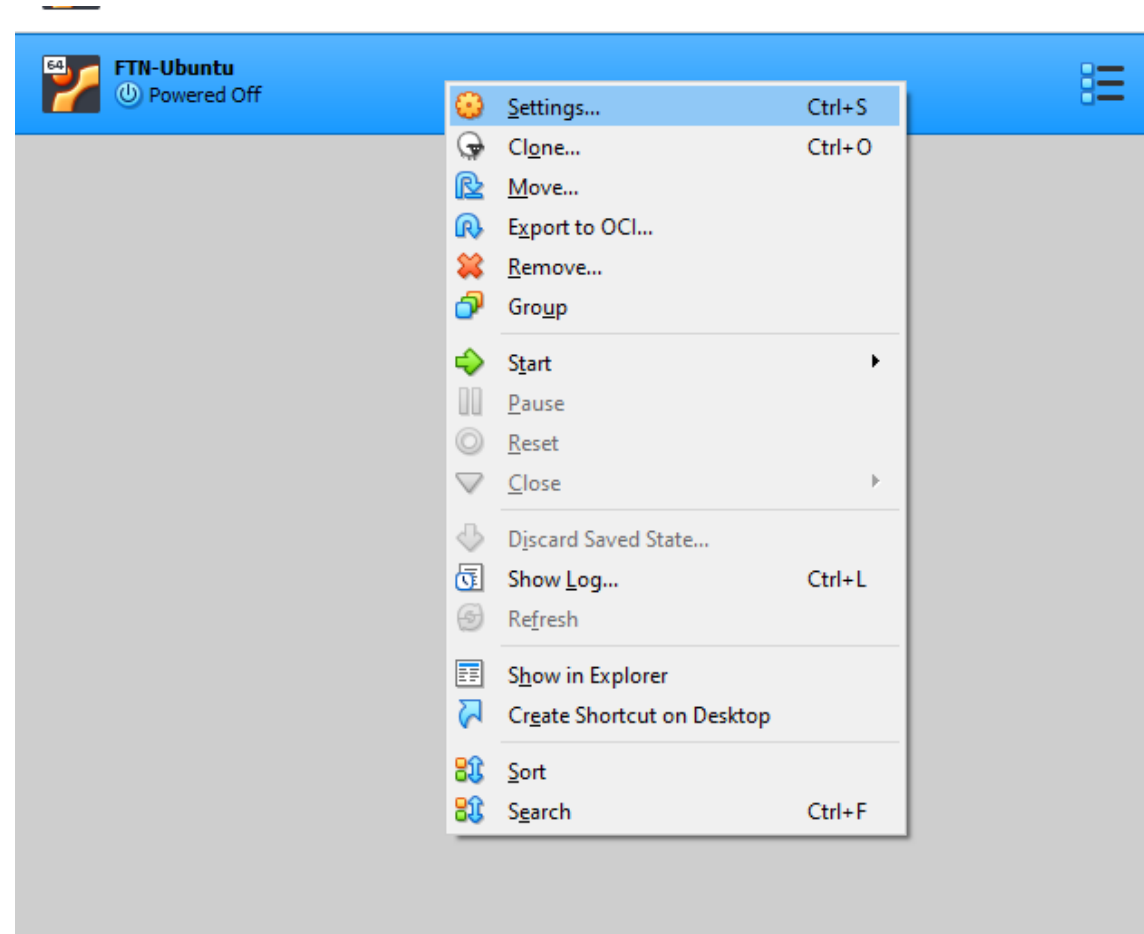
Korak #2 Disk



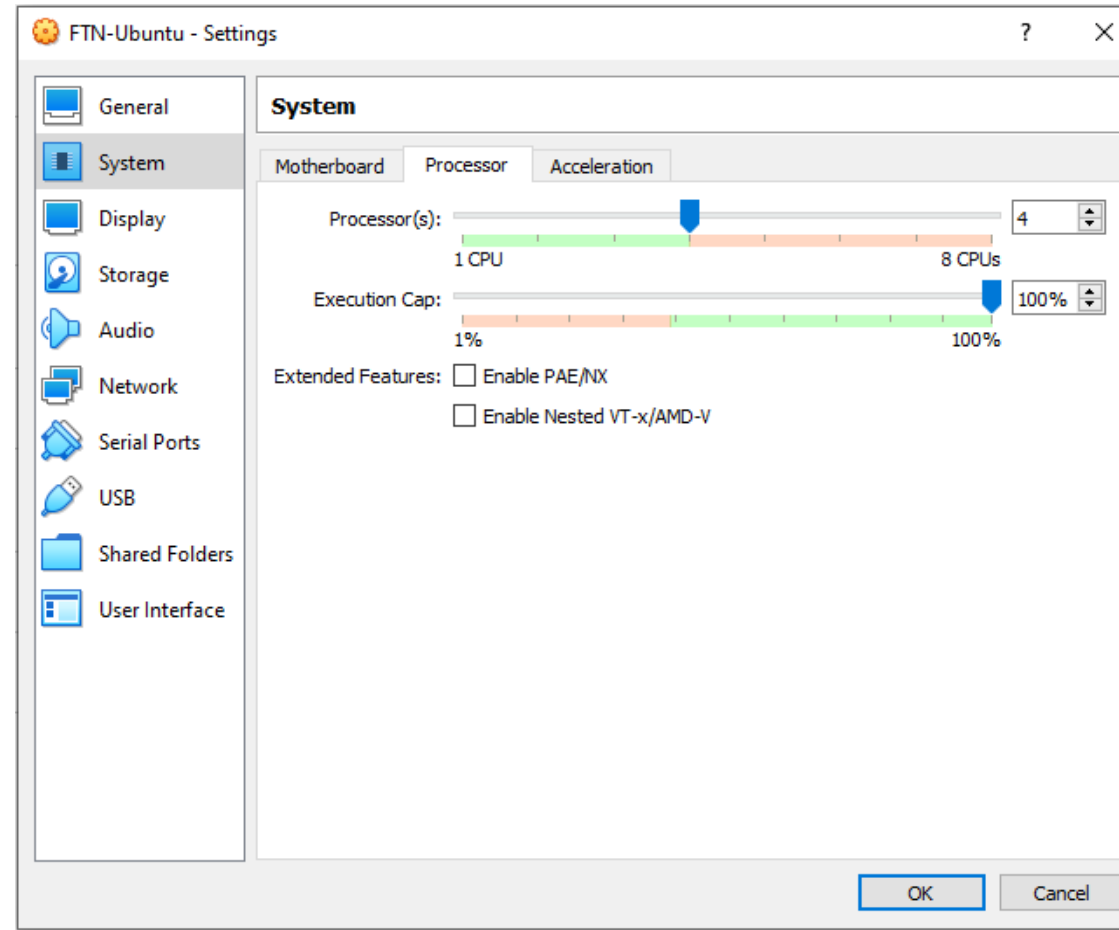
Korak #2 Disk

- Vaš virtuelni računar mora imati hard disk
- Taj virtuelni disk će biti fajl na vašem računaru negde.
- Alocirano je 40GB što je generalno dovoljno za laganu upotrebu. Apsolutni minimum je oko 25GB.
- Ovde je odabrano fiksno alociranje: to znači da taj disk odmah zauzme punih 40GB, ali kao bonus, radi brže. (Ako vas zanima više negde oko nedelje 10 nastave će biti objašnjeno zašto 😊)
- Format nije bitan i može ostati na podrazumevanom osim ako ne planirate da disk koristite i u druge svrhe. Ako to planirate onda sigurno znate i šta vam je potrebno.
- Vodite računa da ova operacija zna da potraje.

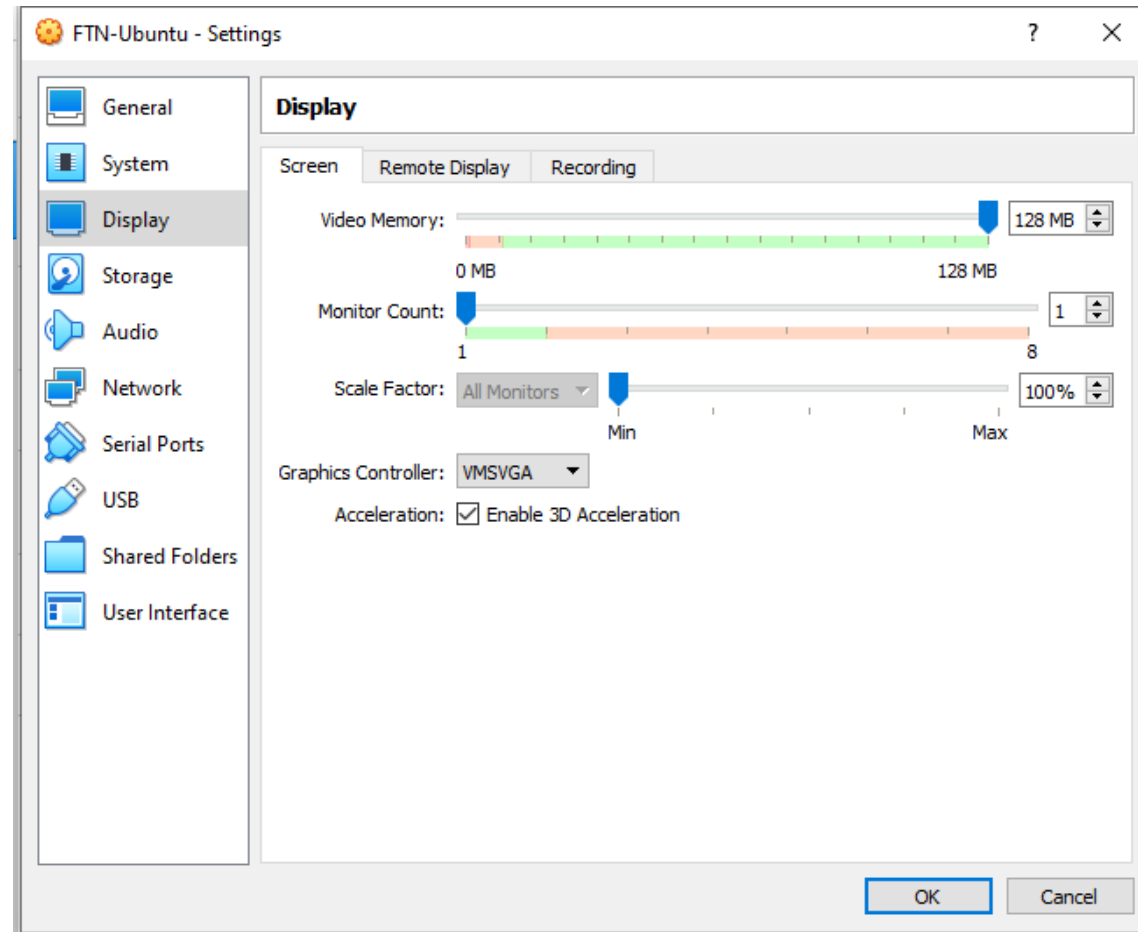
Korak #3: Podešavanje



Korak #3 Podešavanje



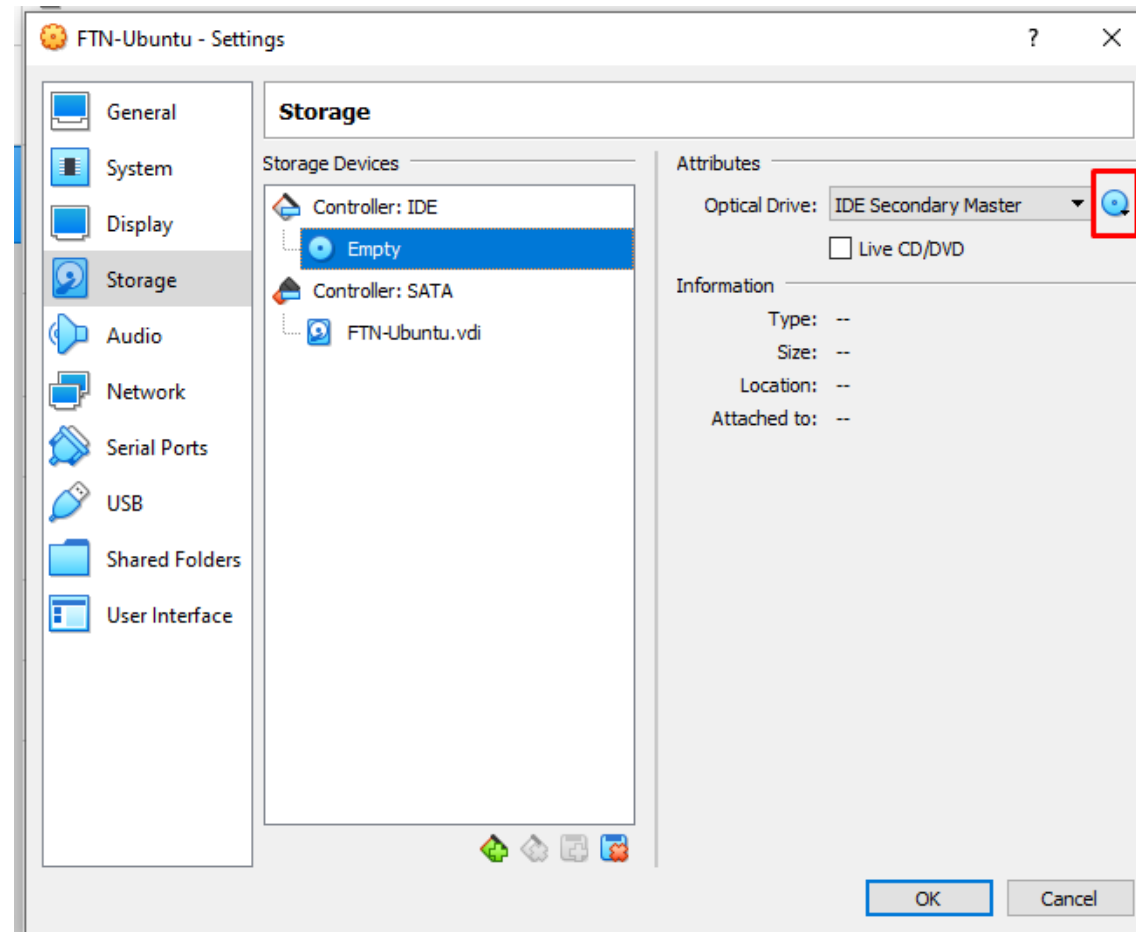
Korak #3 Podešavanje



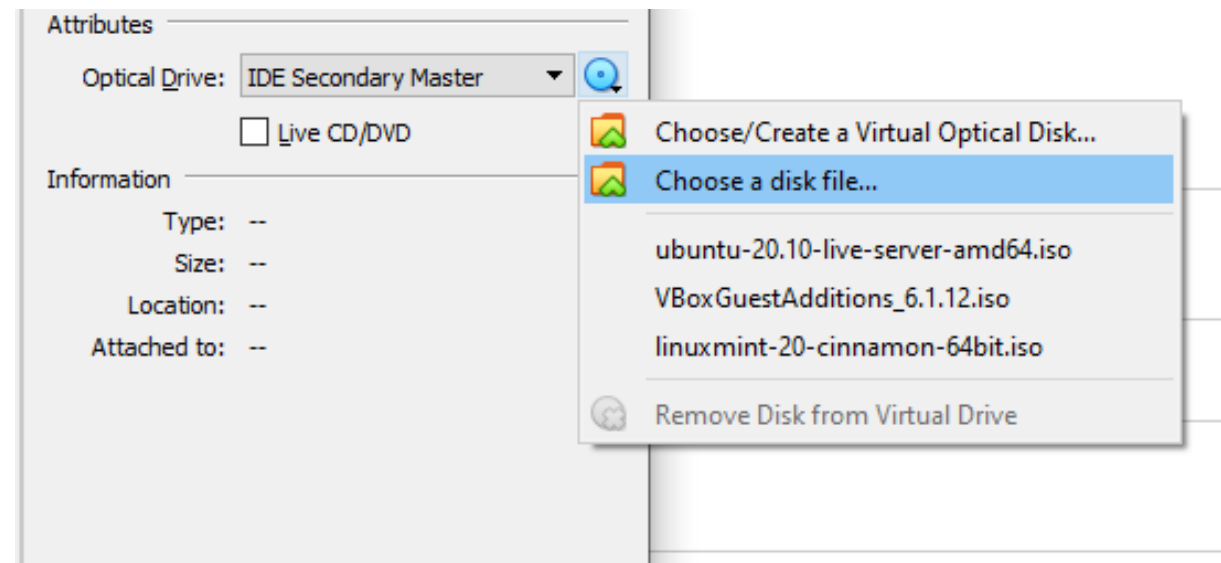
Korak #3 Podešavanje

- Ovde smo uključili da naš VM ima pristup više jezgara procesora (zdravo ako hoćemo da programiramo sa nitima na njemu!)
- Takođe smo poboljšali grafiku tako da prikaz GUI-ja na računaru radi brže i bolje.
- Moguće je koristiti inače VM tako što samo izvršavate kod na Linux-u a koristite ga sa udaljenog računara preko SSH-a i koristite Visual Studio Code editor u udaljenom režimu da editujete fajlove na njemu. Onda mu ne treba nikakav ekran i displej. Podešavanje zahteva malo finese ali radi jako dobro.

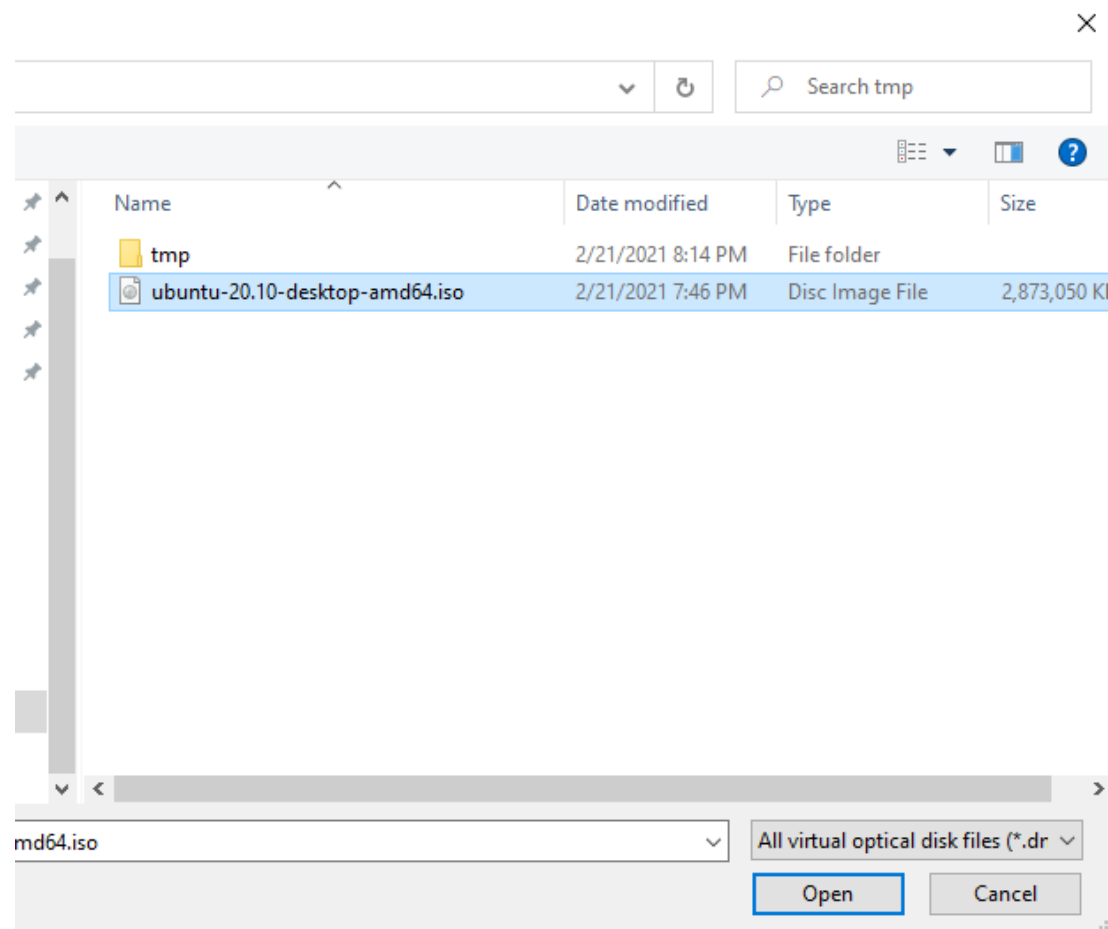
Korak #4 Pokretanje instalacije



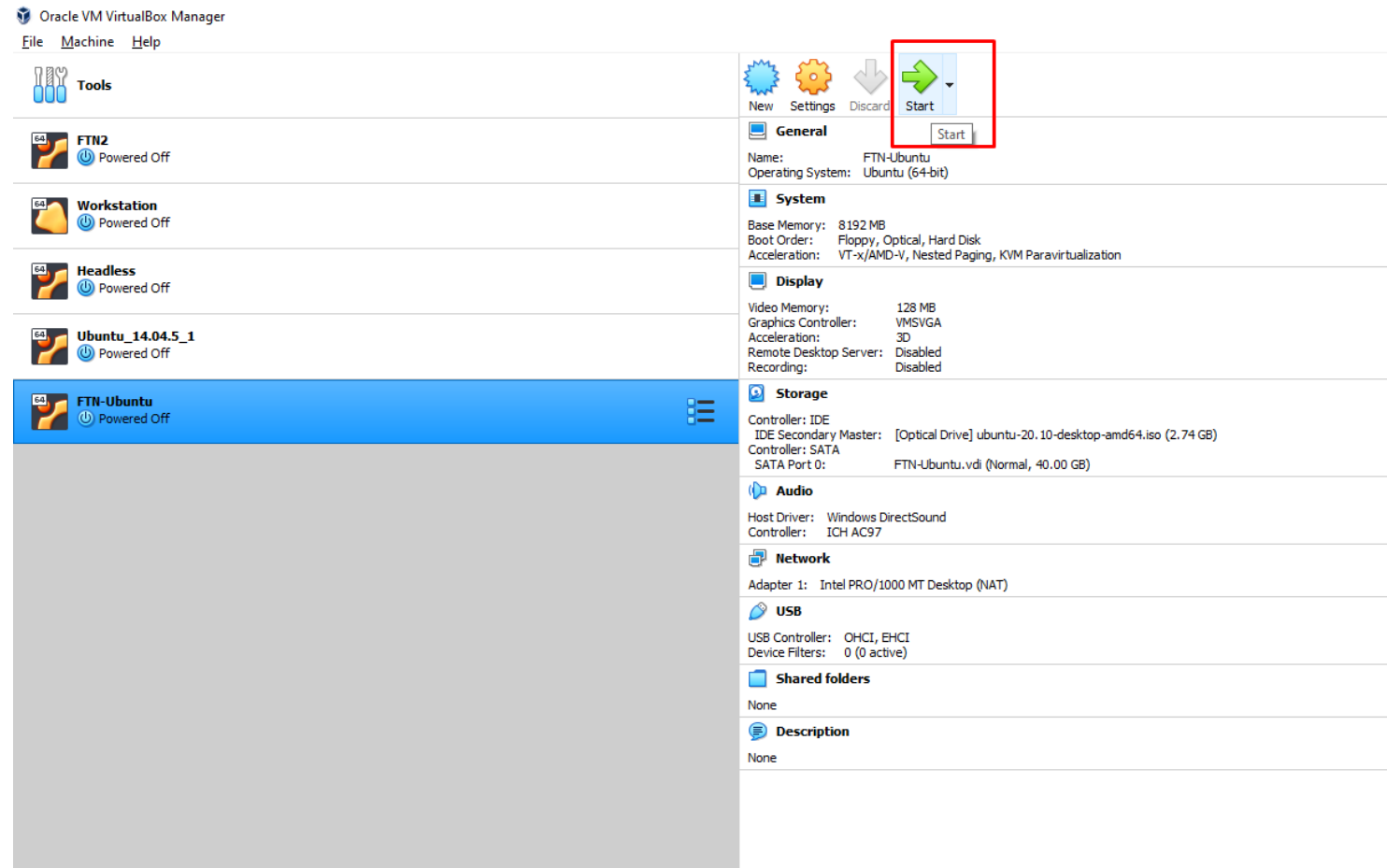
Korak #4 Pokretanje instalacije



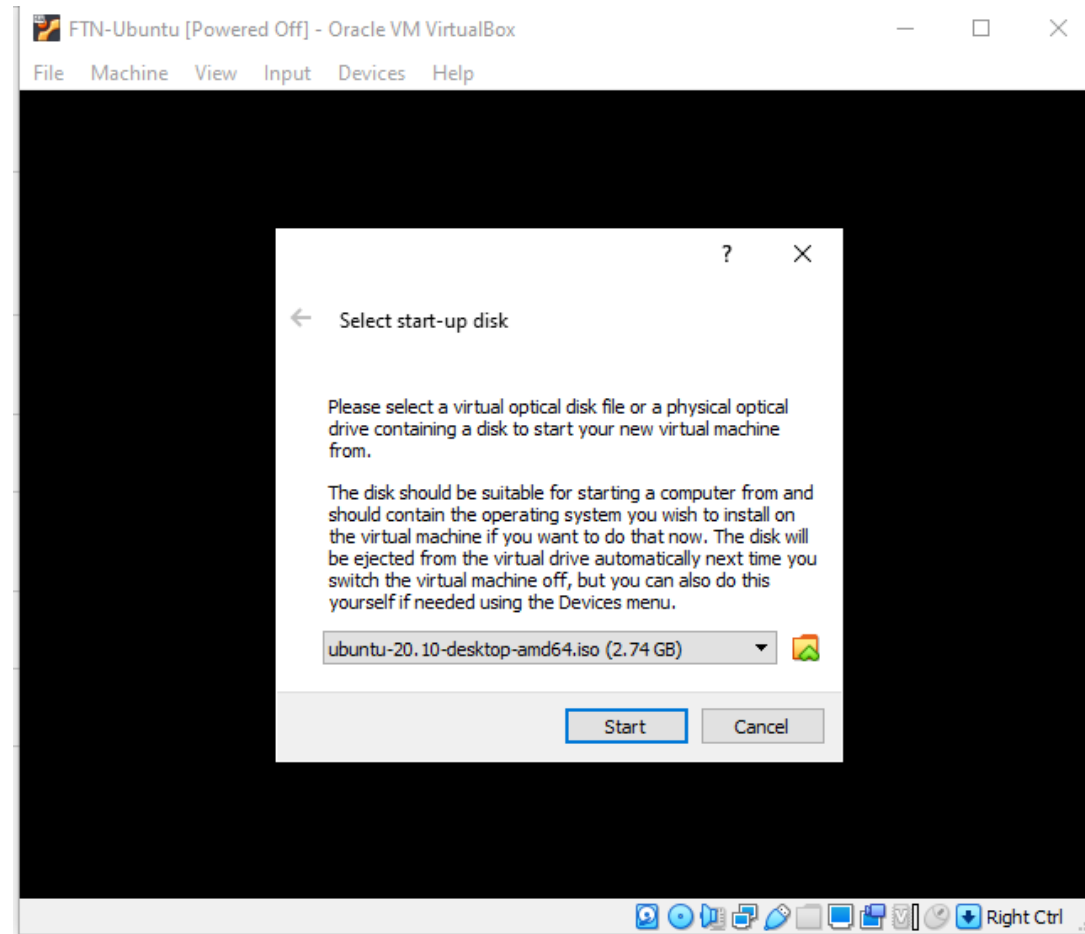
Korak #4 Pokretanje instalacije



Korak #4 Pokretanje instalacije

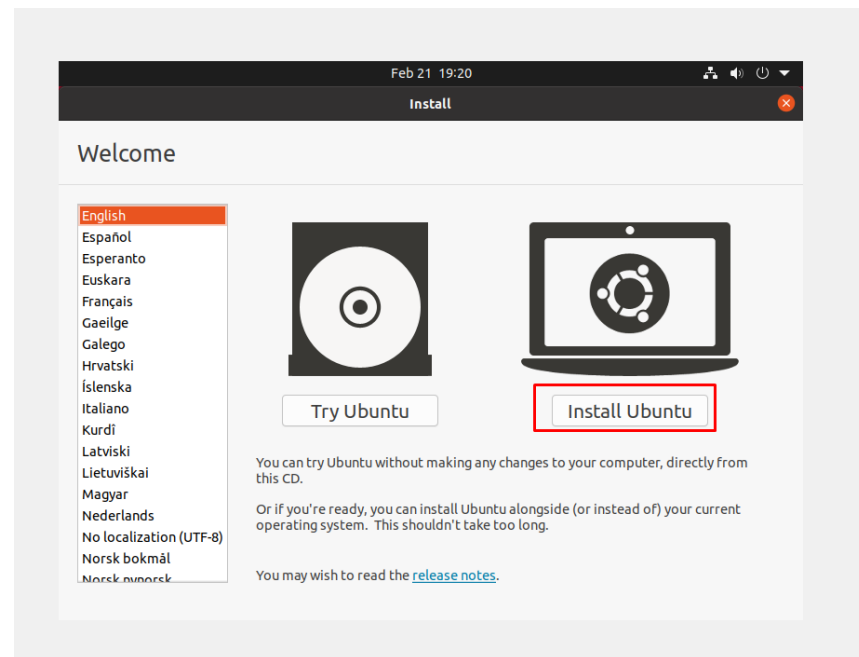


Korak #4 Pokretanje instalacije

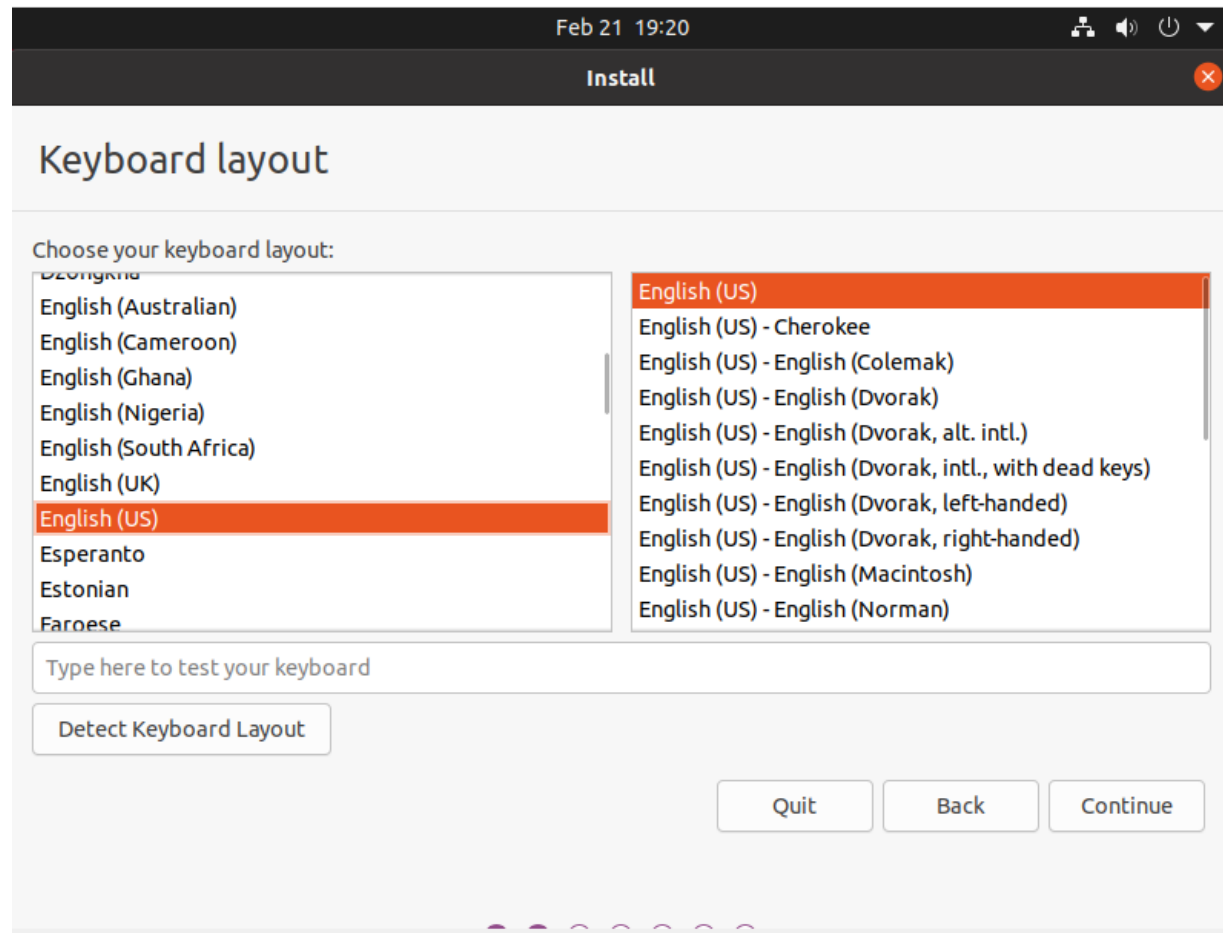


Korak #4 Pokretanje instalacije

- Boot potraje ali kada se završi nalazite se u LiveCD okruženju u kome koristite Ubuntu 'instaliran' na CD-u koji ste ubacili u vaš virtuelni računar.
- Imate izbor da ga probate, ali ono što nama treba je instalacija



Korak #5 Podešavanje instalacije

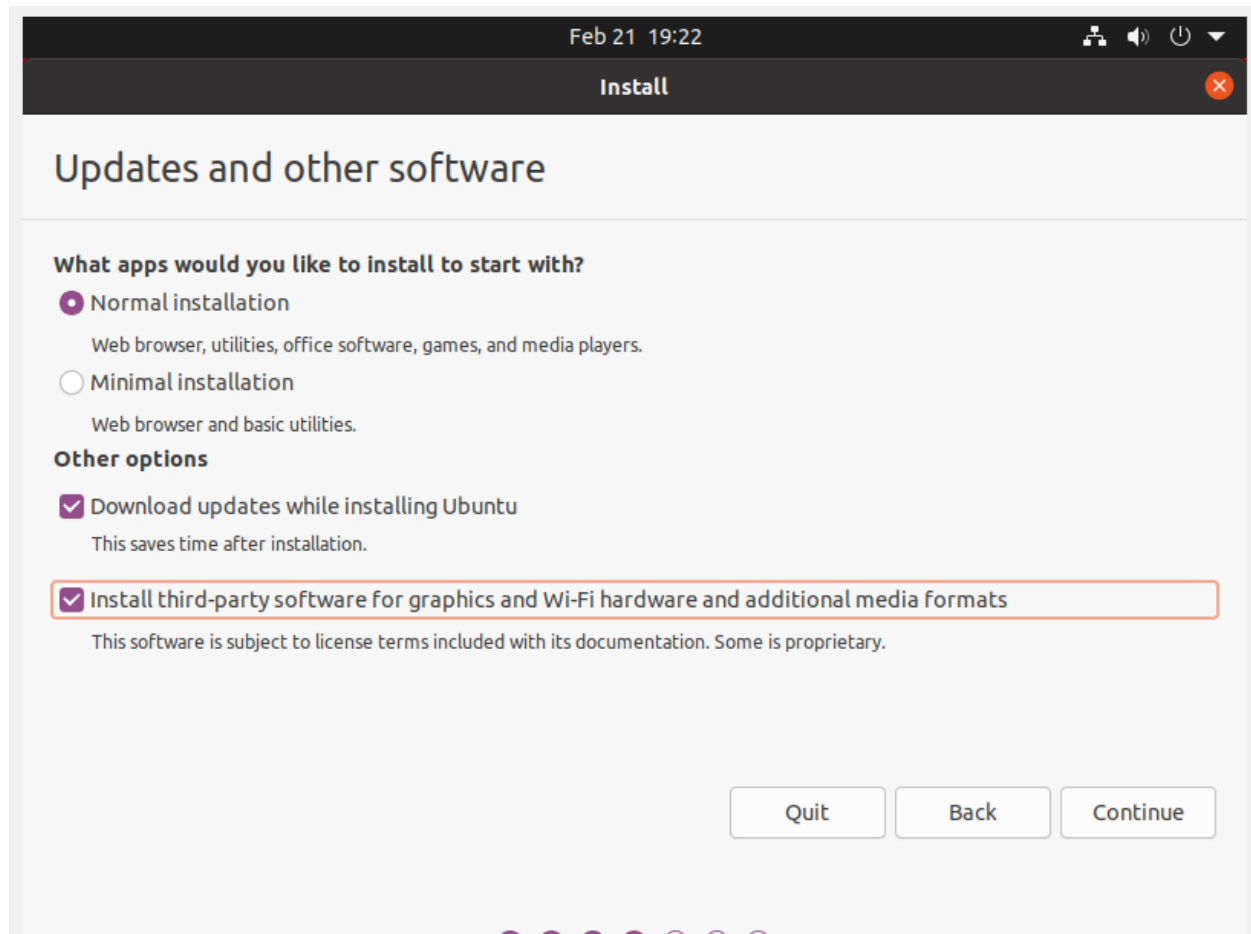


Možete da koristite kakvu god hoćete tastaturu, naravno.

US English je zgodna zato što u ovom sistemu uglavnom programirate, a SR tastatura vam ne da da lako otkucate jako puno dodatnih karaktera.

Naravno, latinicu, ćirilicu, i sve drugo što možete poželeti možete dodatni naknadno u sistem uz odabir layout-a kao i u Windows okruženju.

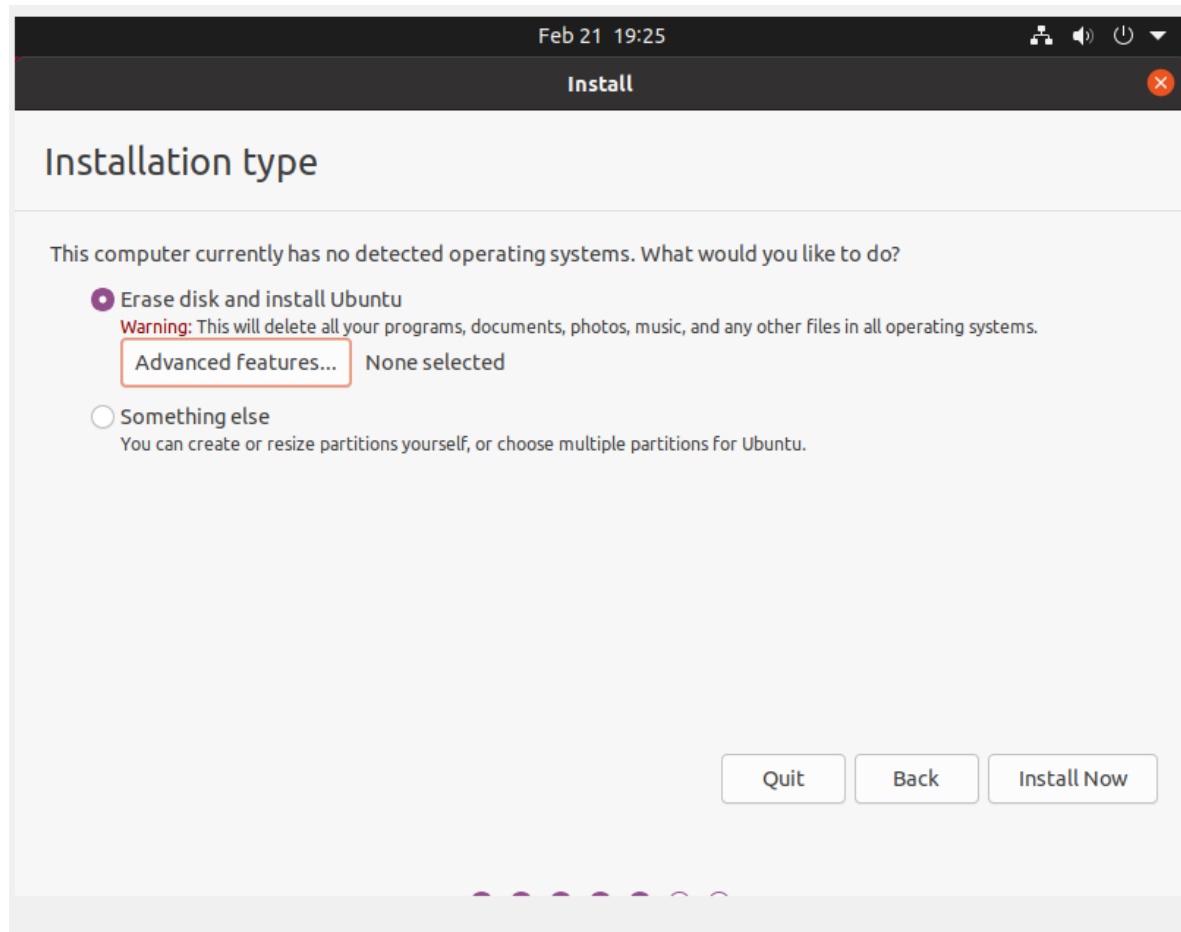
Korak #5 Podešavanje instalacije



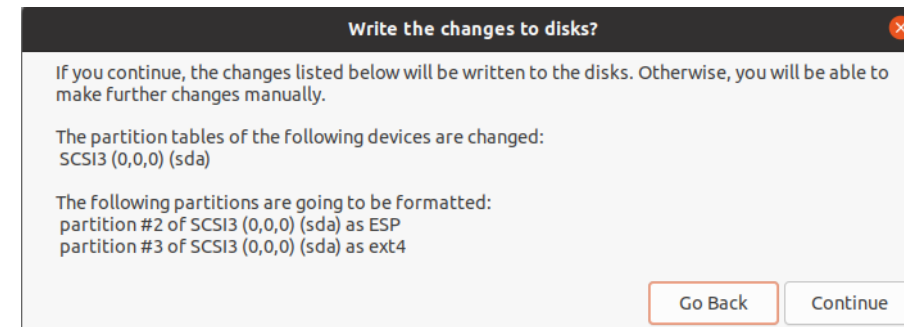
Ovo je gotovo sigurno ono što želite. Ako odaberete minimalnu instalaciju nećete imati gomilu viška softvera, ali prostora ima i može da bude korisno.

Third-party softver se odnosi na drajvere i slične ne Free and Open Source elemente. Za čistunce Linux ideologije možete isključiti ovu opciju, ali može pomoći tu i tamo.

Korak #5 Podešavanje instalacije

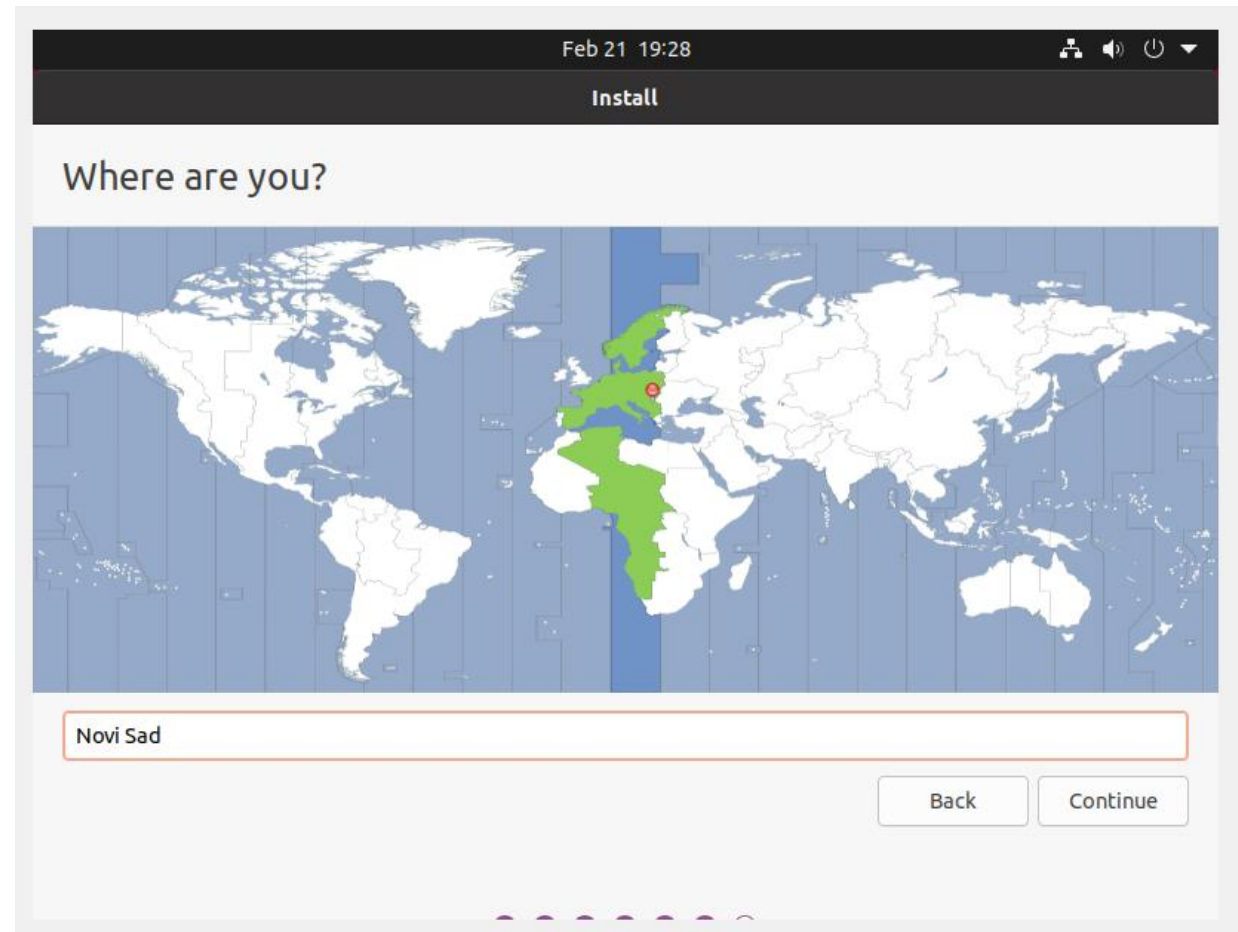


Nema brige: ovo se odnosi na potpuno virtuelni disk koji smo malopre napravili. Pošto vam na njemu ne treba ništa osim Linux-a možete odabrati ovu opciju po automatizmu.

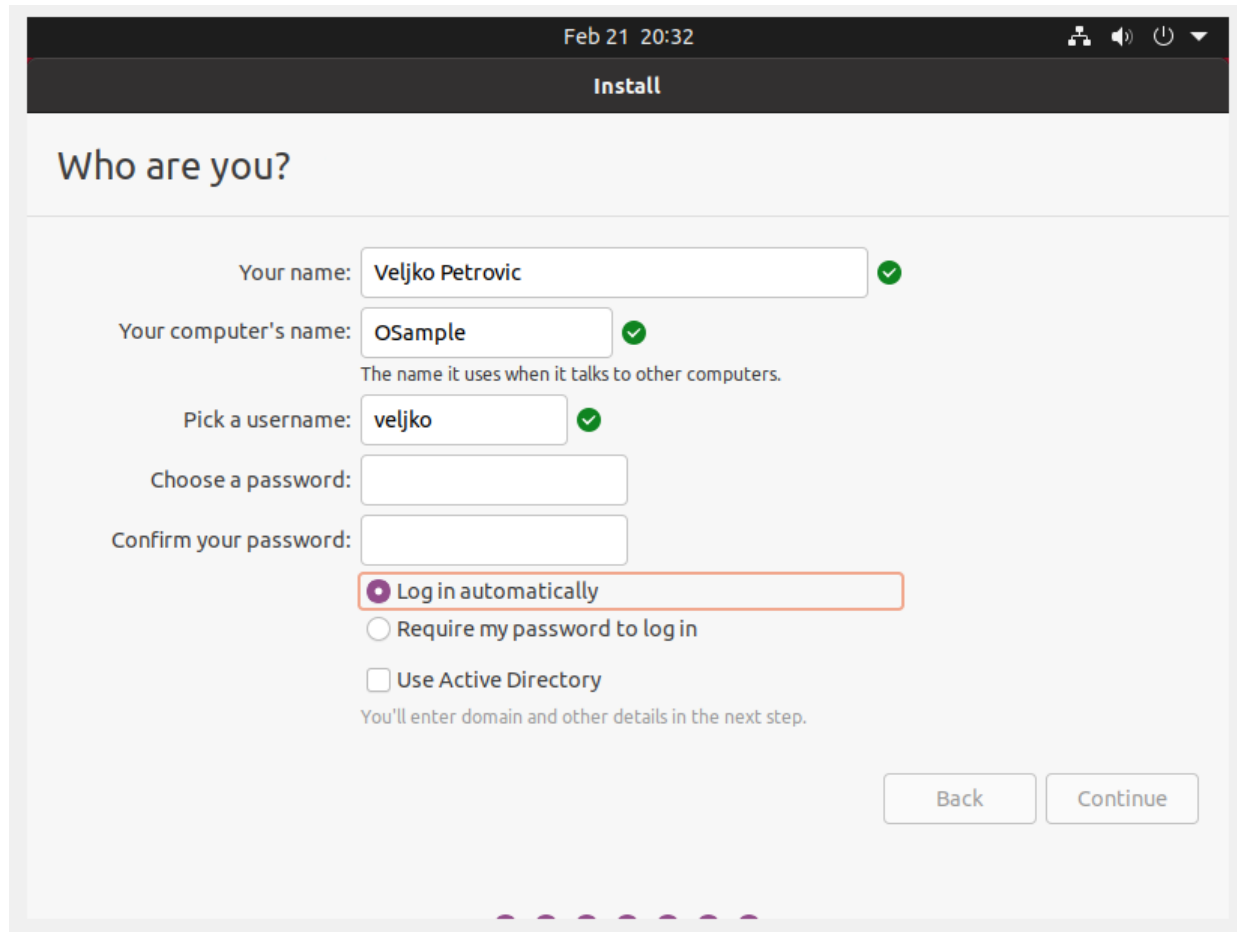


Ne zaboravite da potvrdite promene sa 'Continue'

Korak #5a: Vežba u očiglednosti



Korak #5 Podešavanje Instalacije



The screenshot shows the 'Who are you?' setup screen from a Windows 10 installation. The window title is 'Install' and the system clock shows 'Feb 21 20:32'. The screen contains the following fields and options:

- Your name:** A text box containing 'Veljko Petrovic' with a green checkmark to its right.
- Your computer's name:** A text box containing 'OSample' with a green checkmark to its right. Below this box is the text: 'The name it uses when it talks to other computers.'
- Pick a username:** A text box containing 'veljko' with a green checkmark to its right.
- Choose a password:** An empty password text box.
- Confirm your password:** An empty password text box.
- Login options:** Three radio buttons are present:
 - ☒ Log in automatically (This option is highlighted with a red rectangular border)
 - ☐ Require my password to log in
 - ☐ Use Active DirectoryBelow these options is the text: 'You'll enter domain and other details in the next step.'
- Navigation:** 'Back' and 'Continue' buttons are located at the bottom right of the form area.

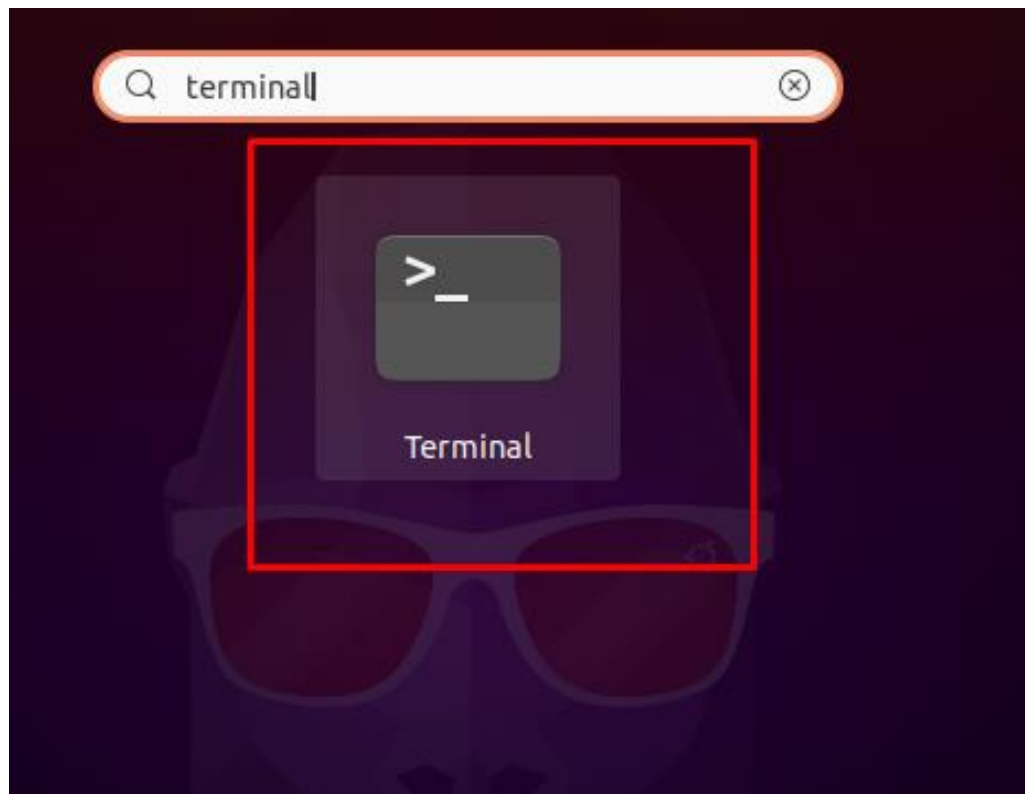
U slučaju povećane paranoje ili deljenog računara, možete tražiti sebi lozinku svaki put kada se logujete, ali generalno nije potrebno.

Nemojte zaboraviti lozinku. 😊

Korak #6: Čekanje...

- Instalacija može da potraje.
- Sačekajte da se završi.

Korak #7 Post-instalacija konfiguracija

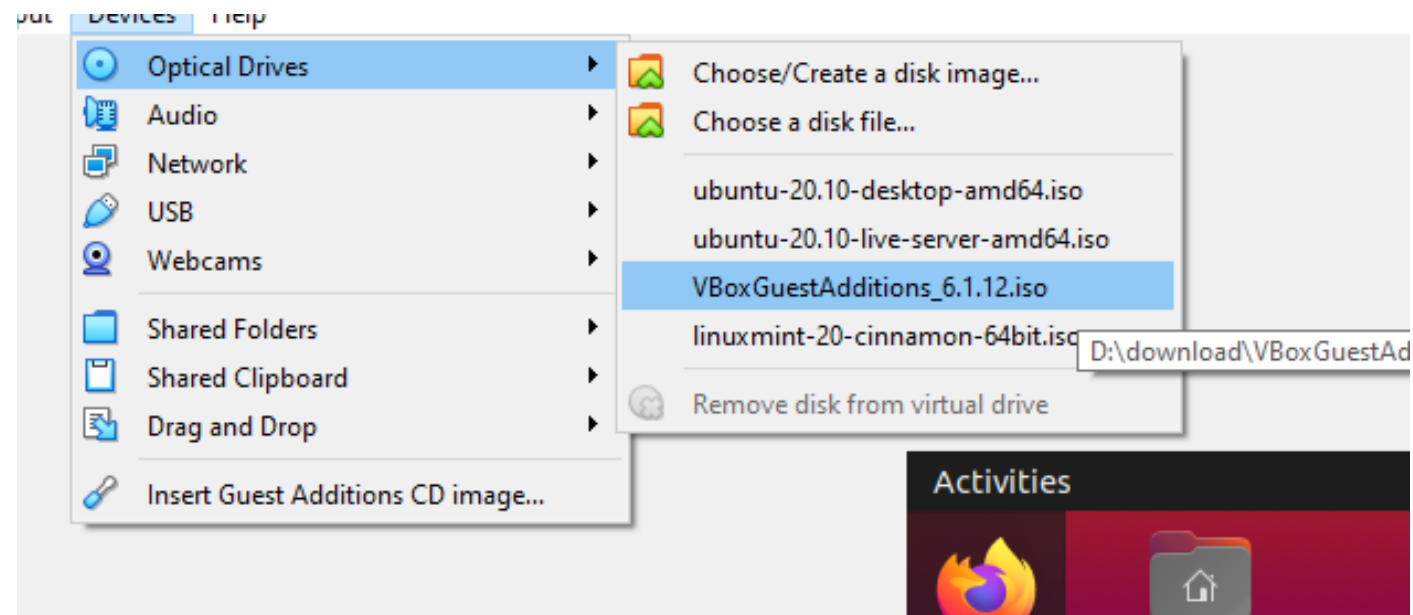


Većinu stvari odradimo u terminalu zato što vam je tako lakše da vidite šta radimo.

Korak #7 Post-instalacija konfiguracija

```
sudo su
apt update
apt upgrade
*Ovde restartovati*
sudo su
apt install build-essential dkms linux-headers-$(uname -r)
```

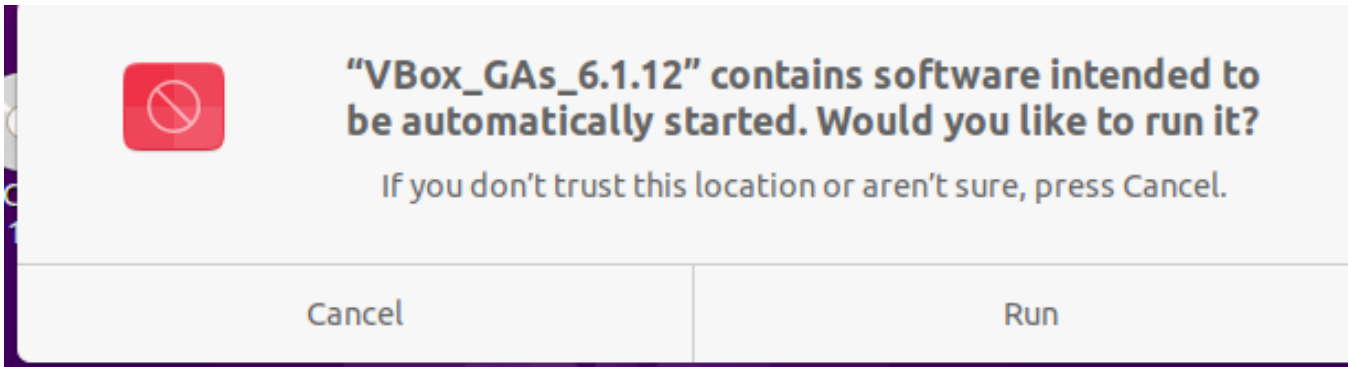
Korak #8: Podešavanje saradnje sa Windows-om



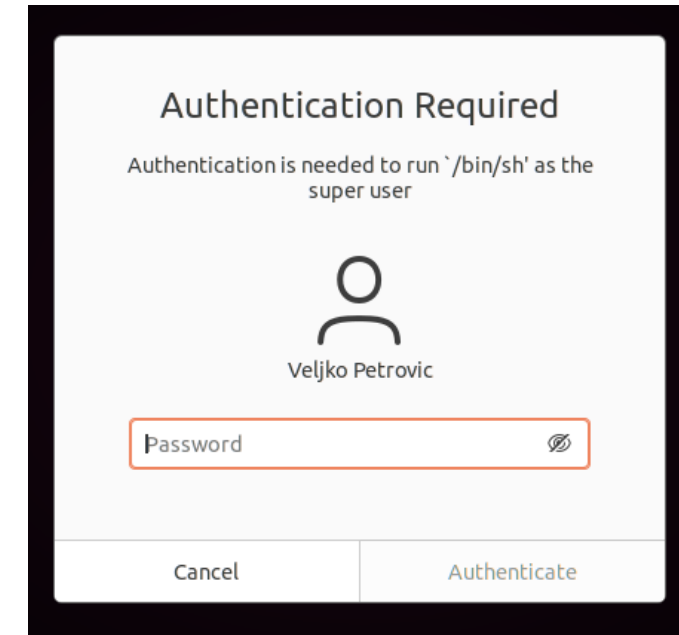
Korak #8: Podešavanje saradnje sa Windows-om

- Ovo je ISO koji na sebi sadrži šta je neophodno da se proširi Linux sa funkcionalnošću da sarađuje u potpunosti sa vašim operativnim sistemom.
- Skidate ga sa istog mesta kao i VirtualBox
- *Uvek* uzmite istu verziju kao i VirtualBox
- Možete koristiti i 'Insert Guest Additions CD' komandu za isti efekat ako nemate pri ruci ISO.

Korak #8: Podešavanje saradnje sa Windows-om

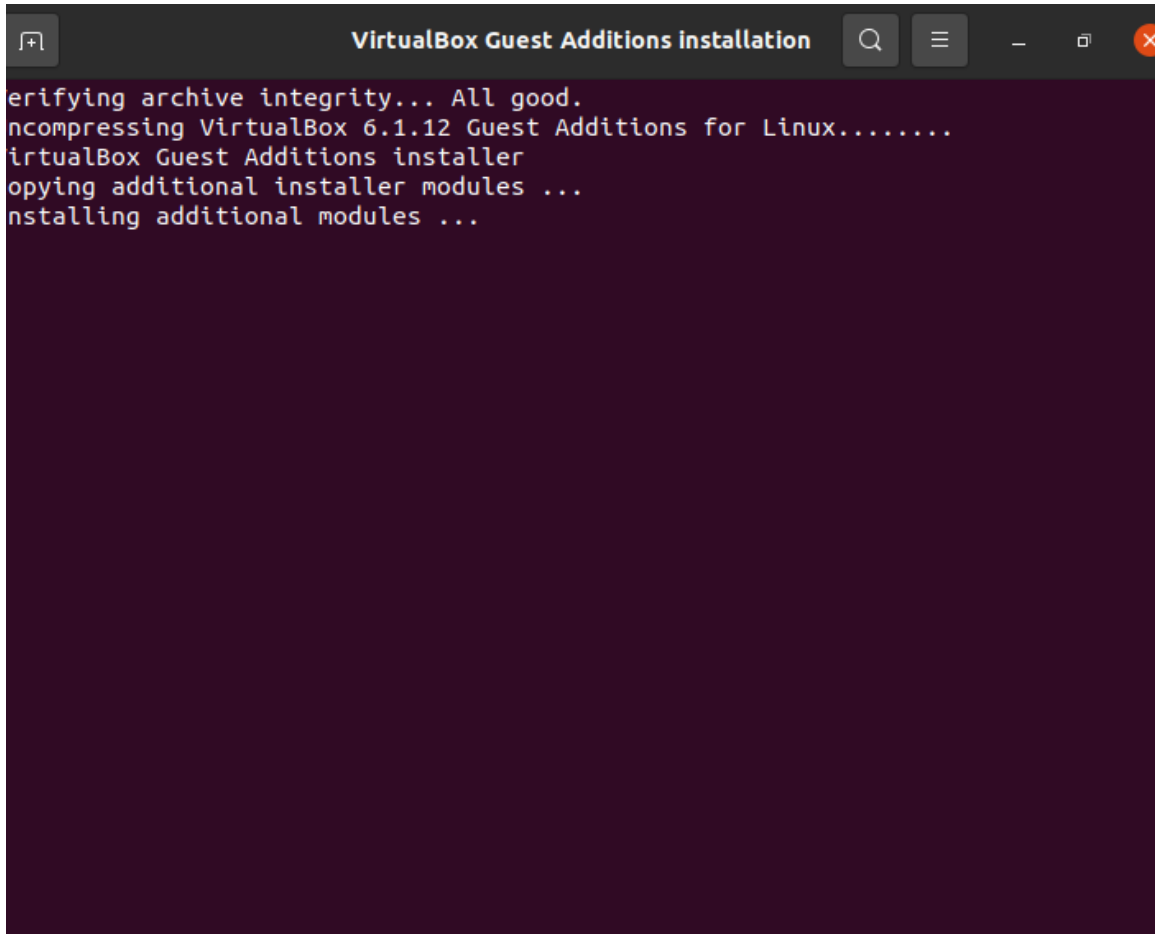


Ovde se iskoristi autorun i dobije se...



Nadam se da niste zaboravili onu lozinku...

Korak #8: Podešavanje saradnje sa Windows-om



```
VirtualBox Guest Additions installation
Verifying archive integrity... All good.
Decompressing VirtualBox 6.1.12 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
```

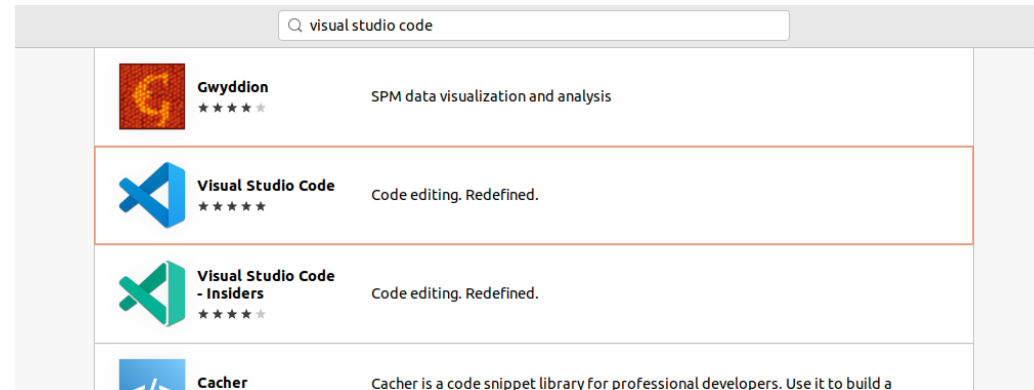
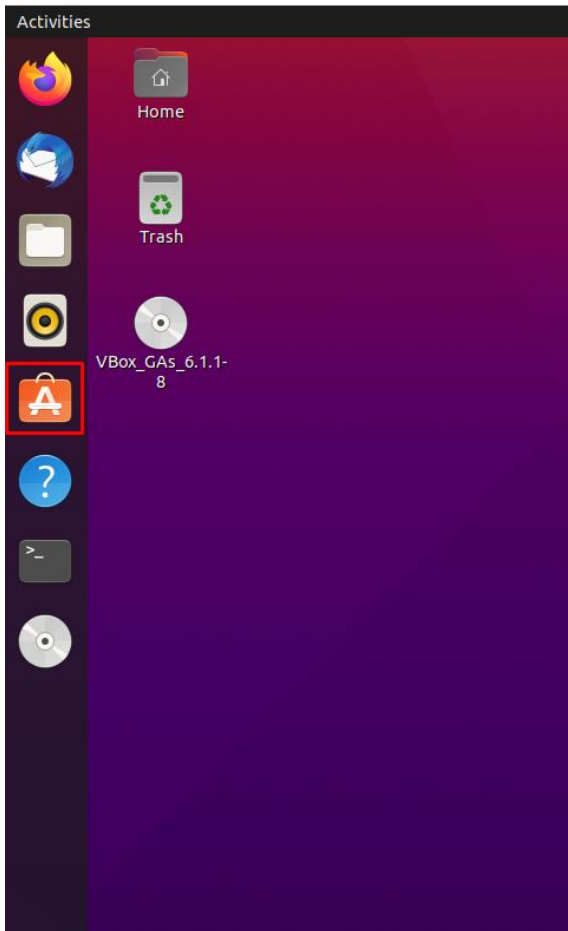
Kad se završi, pritisnite enter i restartujte sistem.

Opcije u Devices o deljenom clipboard-u i drag and drop-u bi trebale da rade, makar donekle. Bagovi su mogući budući da je potpuna integracija teška.

Korak #9 Finalna podešavanja

- Možda ste primetili da je displej jako mali i nezgodan: full-screen opcija rešava taj problem elegantno.
- Možete takođe primenom seamless moda ili podešavanjem rezolucije da podesite da vam prozor bude odgovarajućih dimenzija.
- Sve što vam je ostalo za podešavanje jeste da instalirate okruženje za programiranje.
- Opcije koje preporučujemo su Code:Blocks i Visual Studio Code. VS Code je moćno rešenje koje se može koristiti za manje više bilo koji programski jezik ikada dok je Code:Blocks fokusiraniji i malo lakši za podešavanje.
- Ako koristite VS Code vodite računa da su mu potrebne ekstenzije da bi doživeo svoj vrhunac.
- Ako ste malo iskusniji možete koristiti šta god da vam je omiljeni editor i konzolu.

Korak #9 Finalna podešavanja



```
sudo apt install codeblocks
```

Korak #10 Testiranje

- Sada ono što moramo da uradimo jeste da da testiramo da li sve ovo radi ono što hoćemo, tj. da li će da kompajlira uspešno C++ programe sa nitima.
- Dok to budemo testirali naučićemo protokol za kompajliranje jednostavnih C++ programa.
- Ovo vam omogućava da kompajlirate dati cpp kod rukom bez potrebe za projektima, ili bilo kakvim podešavanjima.

Korak #10 Testiranje – Test kod

```
1  #include <iostream>
2  #include <thread>
3
4  using namespace std;
5
6  void f(){
7      for(int i = 0; i < 10; i++){
8          cout << i << endl;
9      }
10 }
11
12 int main(){
13     thread t1(f);
14     thread t2(f);
15     t1.join();
16     t2.join();
17     return 0;
18 }
```

Nije bitno šta kod radi sada. Ono što je bitno da koristi neke od funkcija koje su nama neophodne.

Korak #10 Testiranje – Kompajliranje

```
g++ --std=c++20 -pthread -o main main.cpp
```

`g++` je ime našeg kompajlera

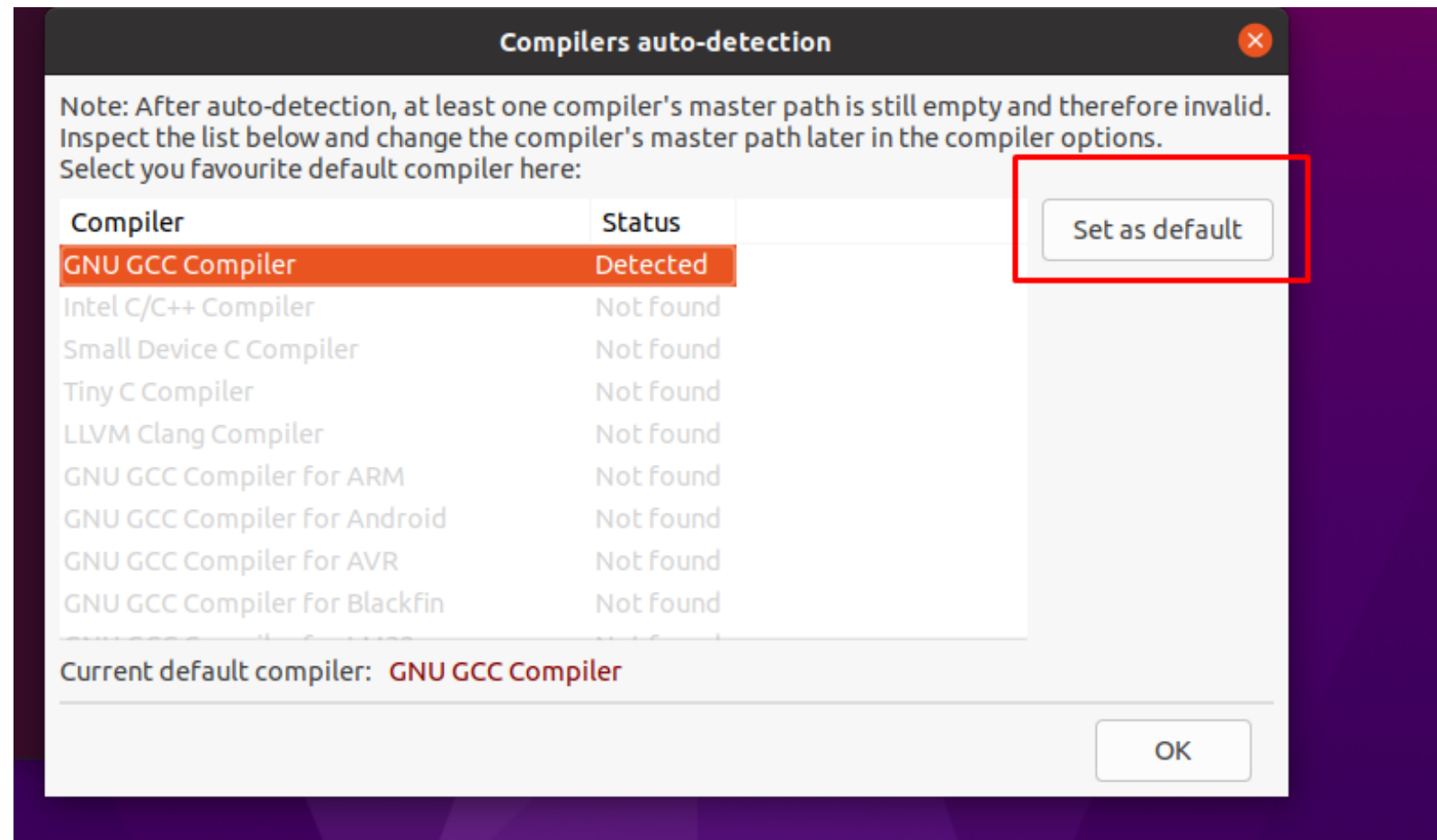
`--std=c++20` je instrukcija da koristimo C++ standard iz 2020. Nije neophodna za ovo, ali testira podršku.

`-pthread` kaže da se koristi pthread biblioteka da se ostvari podrška nitima.

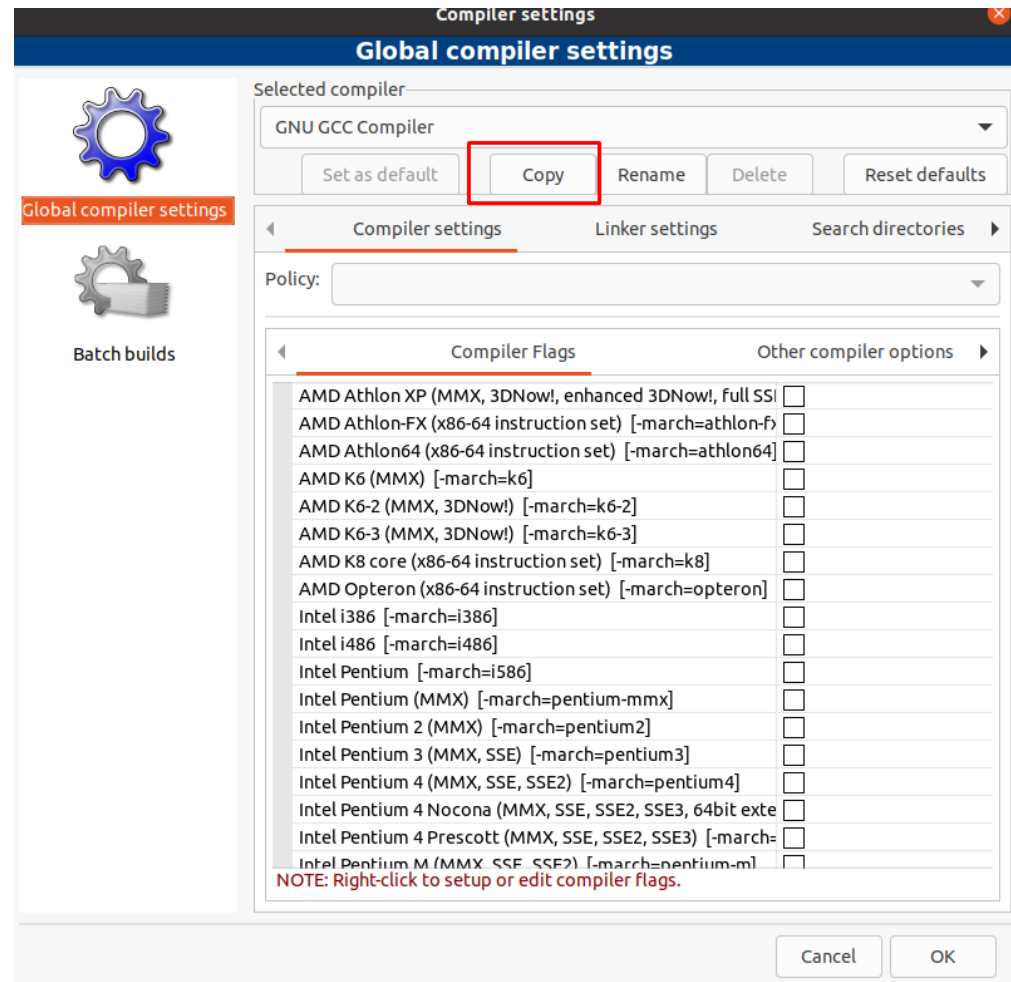
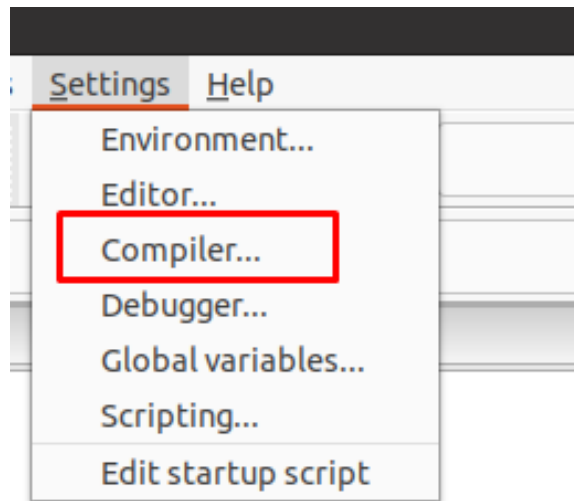
`-o` opcija nam kaže kako se zove izlazni fajl

I konačno `main.cpp` je ono što kompajliramo.

Code:Blocks test

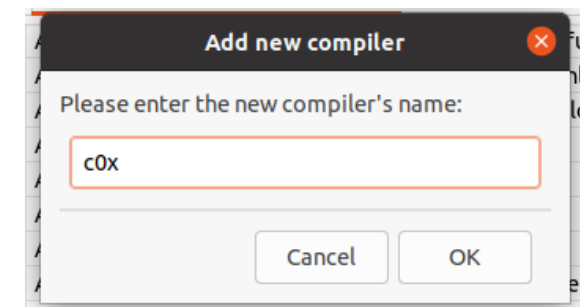


Code:Blocks test

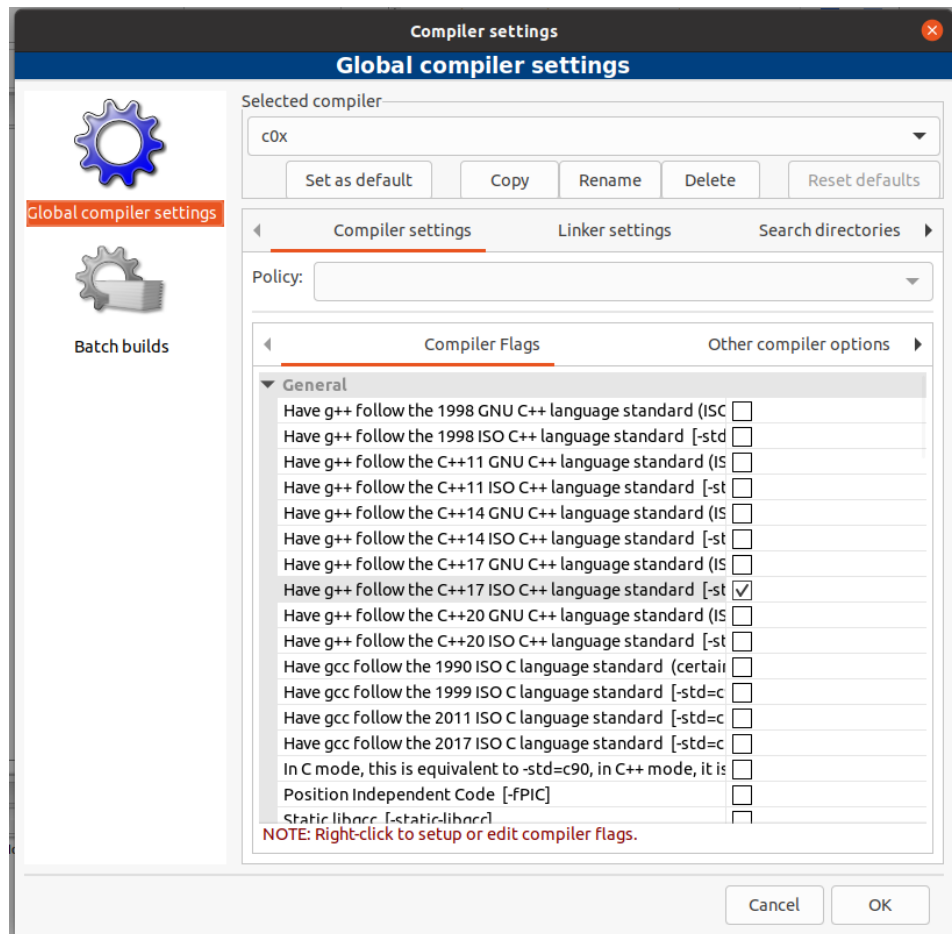


Kopiramo kompajler da bi na jednom mesto podesili sve što nam treba i imali tu posebnu konfiguraciju samo za potrebe ovog predmeta.

Plus, čini stare code blocks projekte kompatibilnim što je bonus.



Code:Blocks test



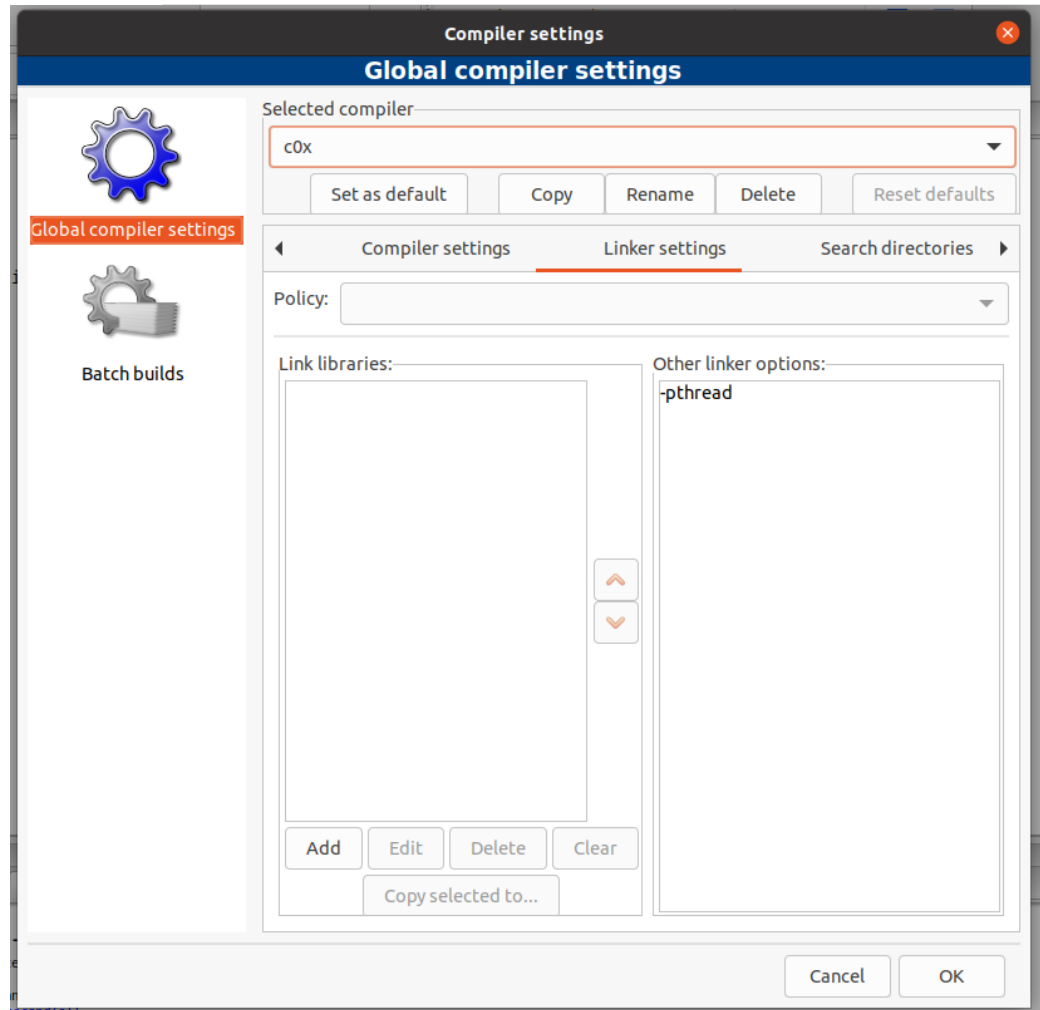
1.

Have g++ follow the C++17 GNU C++ language standard (ISO C++17)	<input type="checkbox"/>
Have g++ follow the C++17 ISO C++ language standard [-std=c++17]	<input checked="" type="checkbox"/>
Have g++ follow the C++20 GNU C++ language standard (ISO C++20)	<input type="checkbox"/>
Have g++ follow the C++20 ISO C++ language standard [-std=c++20]	<input type="checkbox"/>

2.

Linker settings

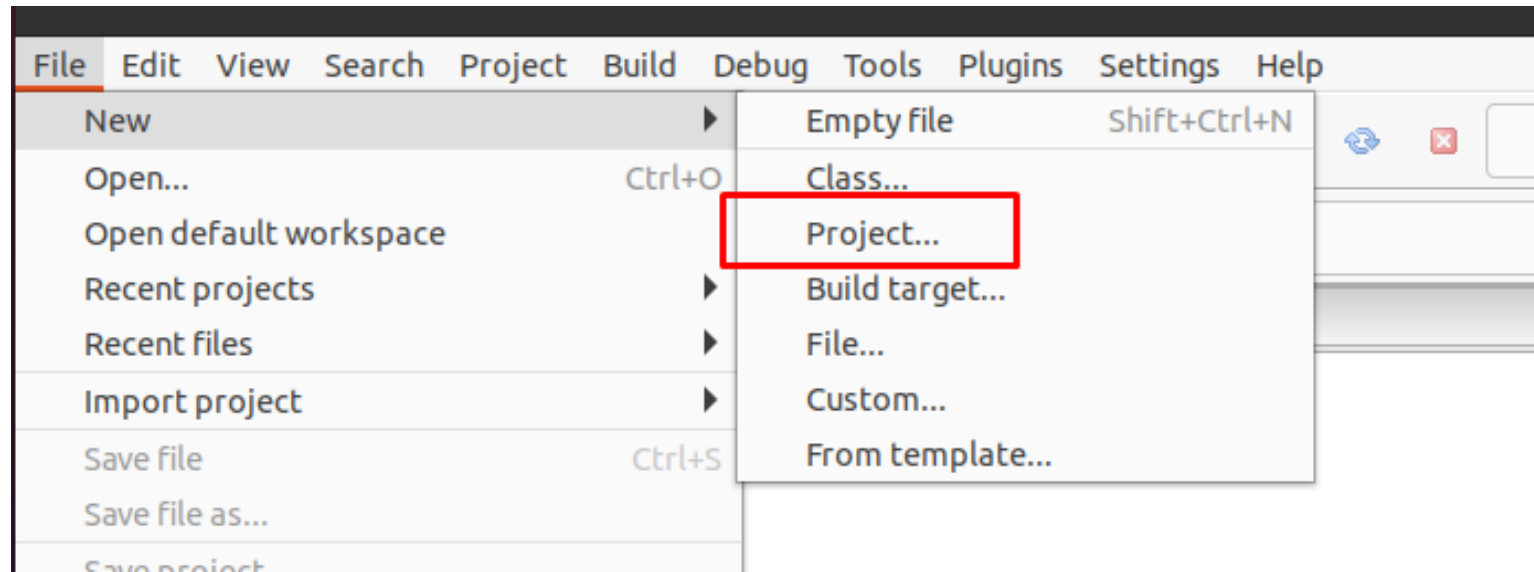
Code:Blocks test



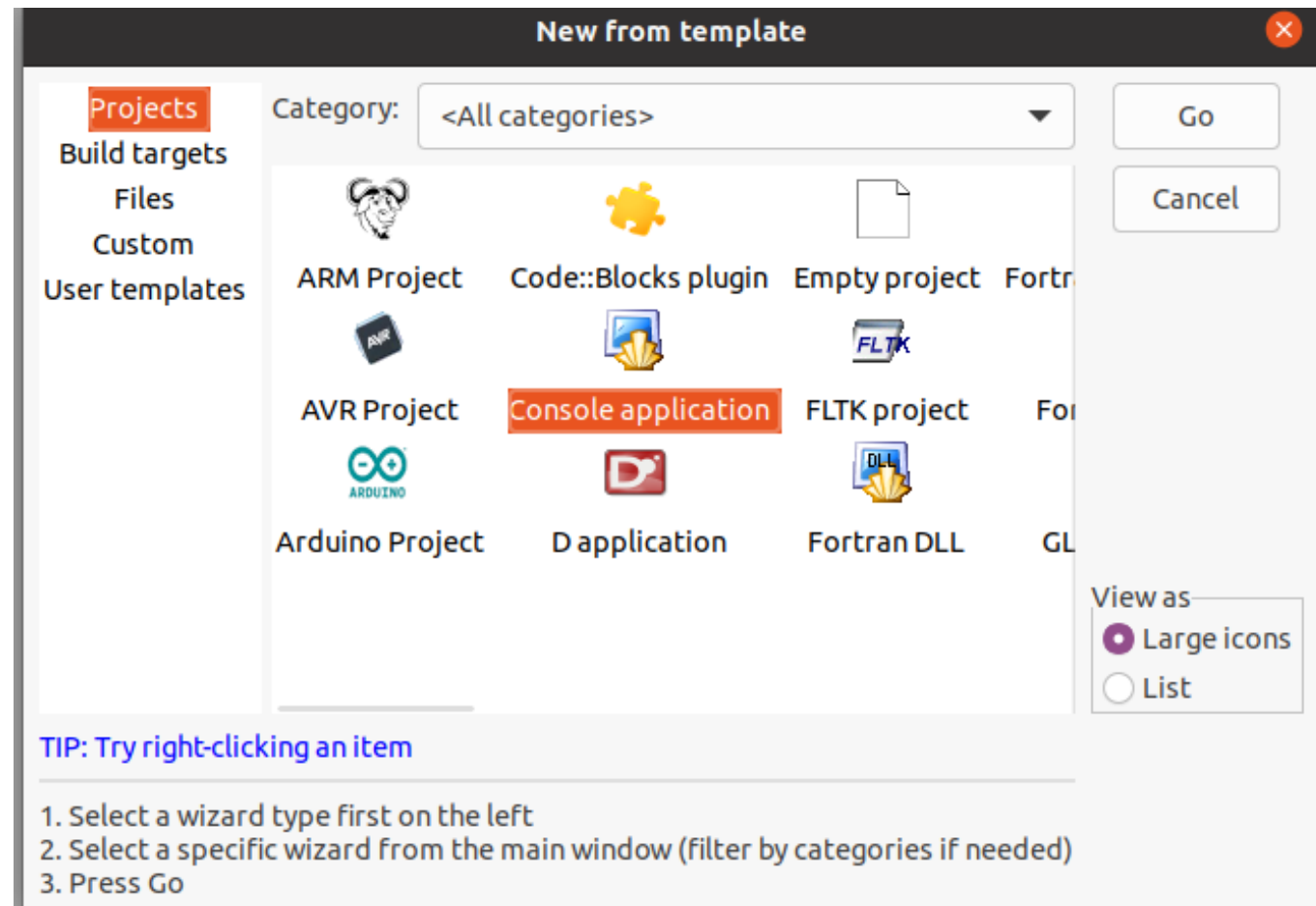
Other linker options: _____

-pthread

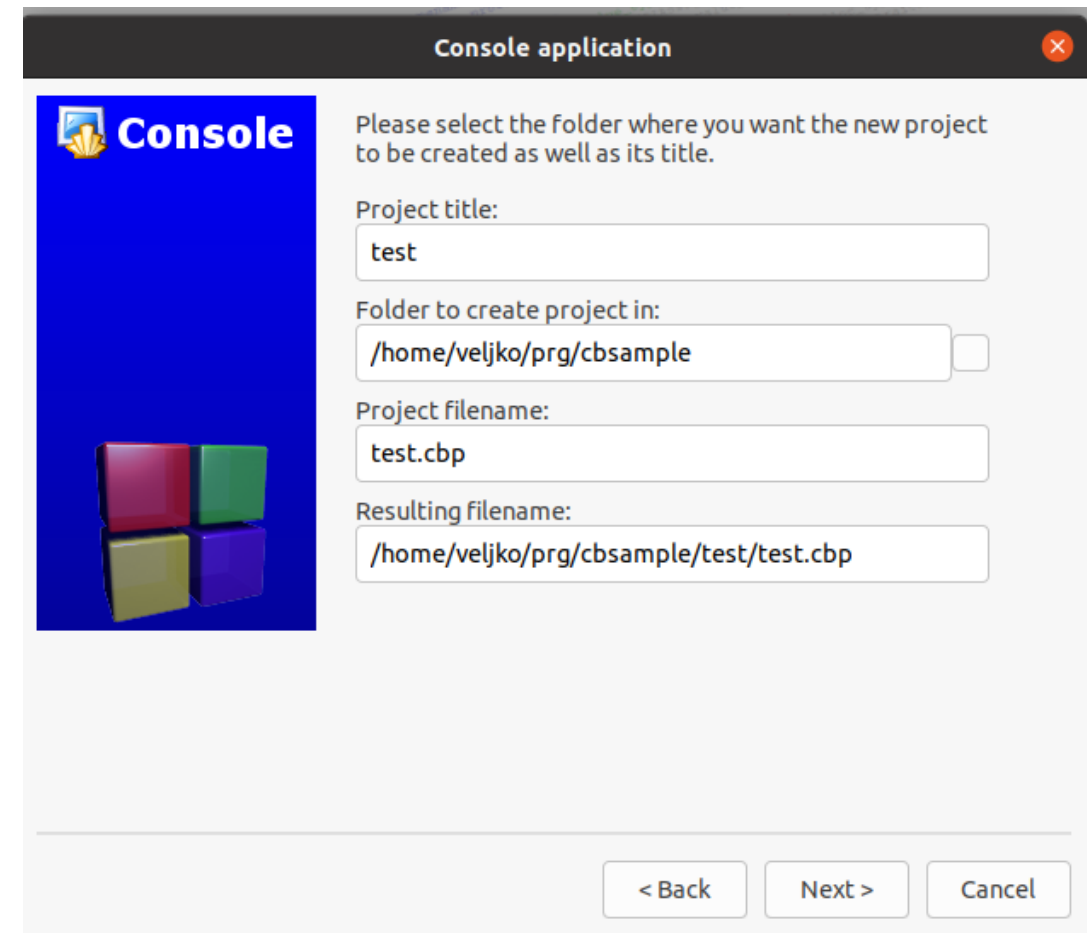
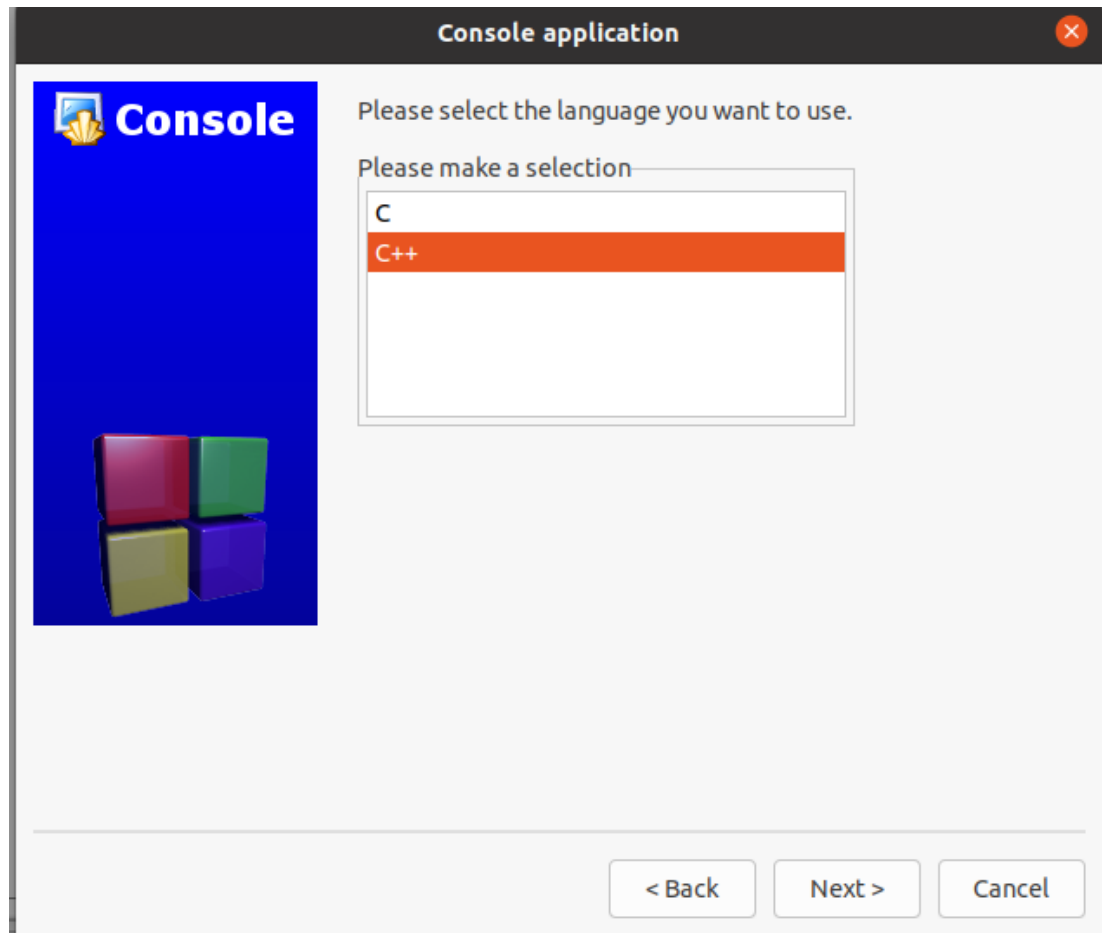
Code:Blocks test



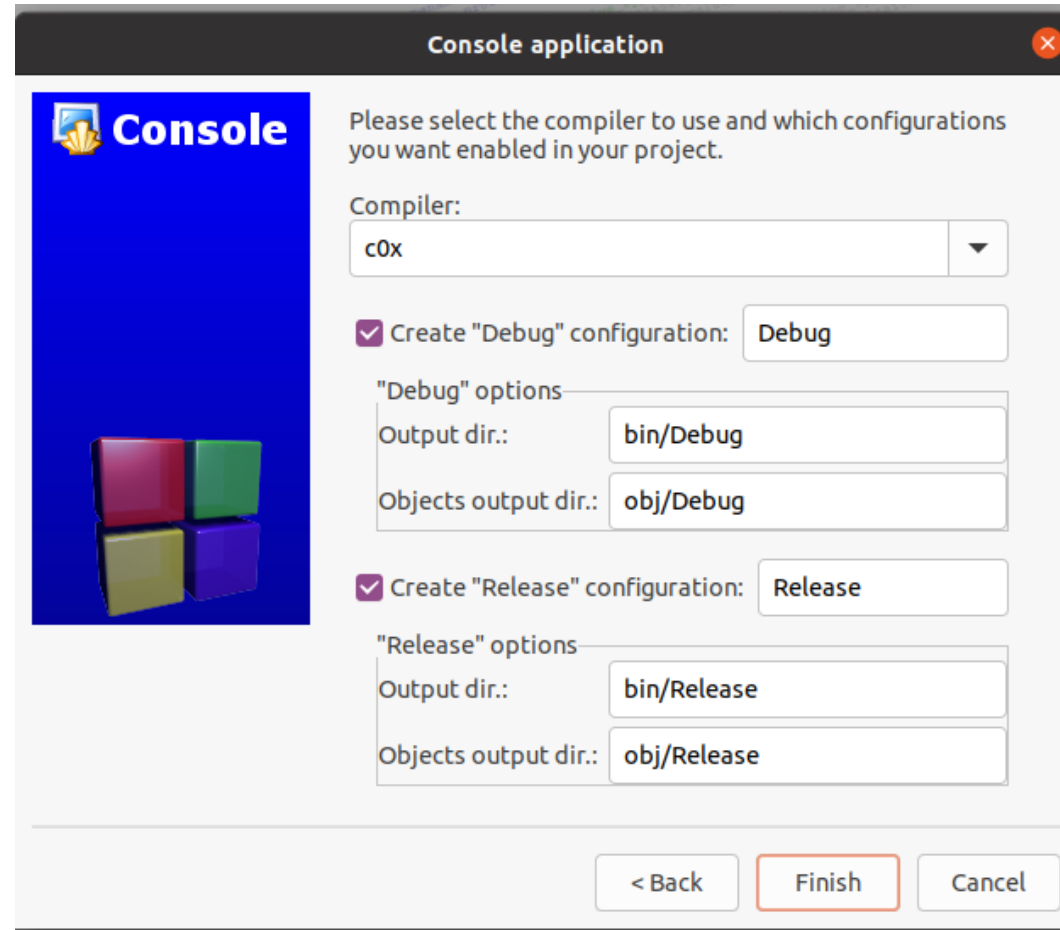
Code:Blocks test



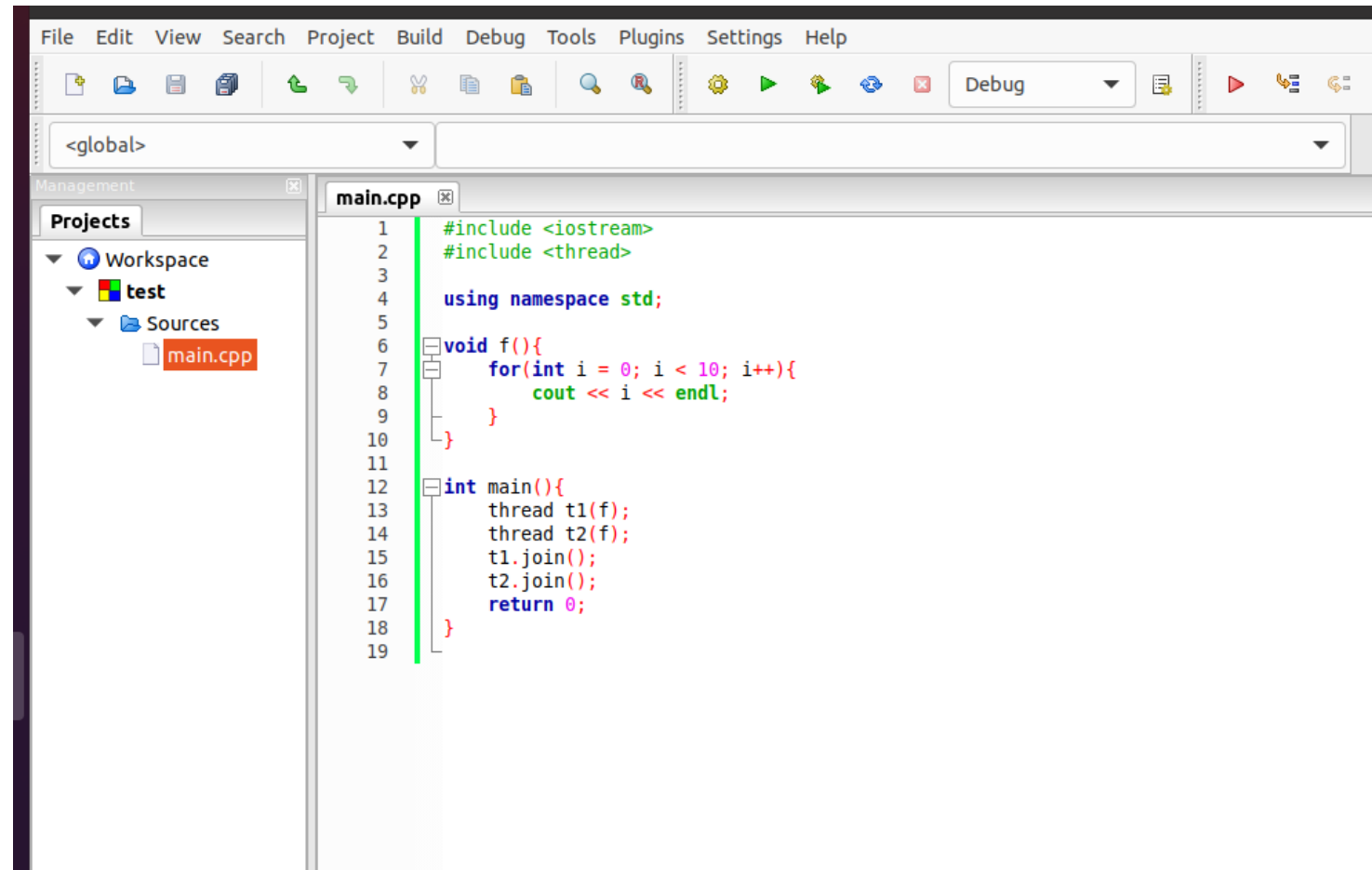
Code:Blocks test



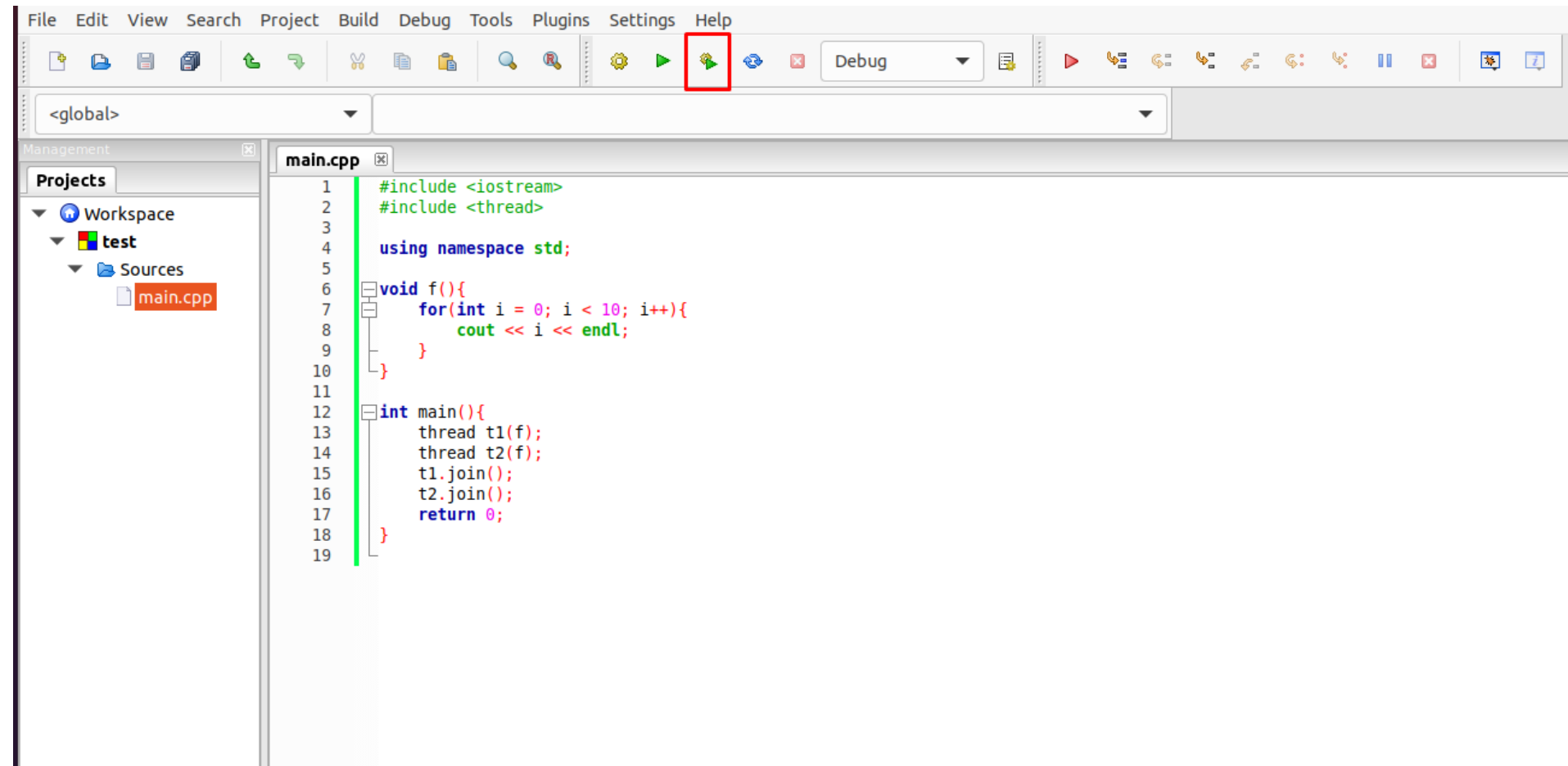
Code:Blocks test



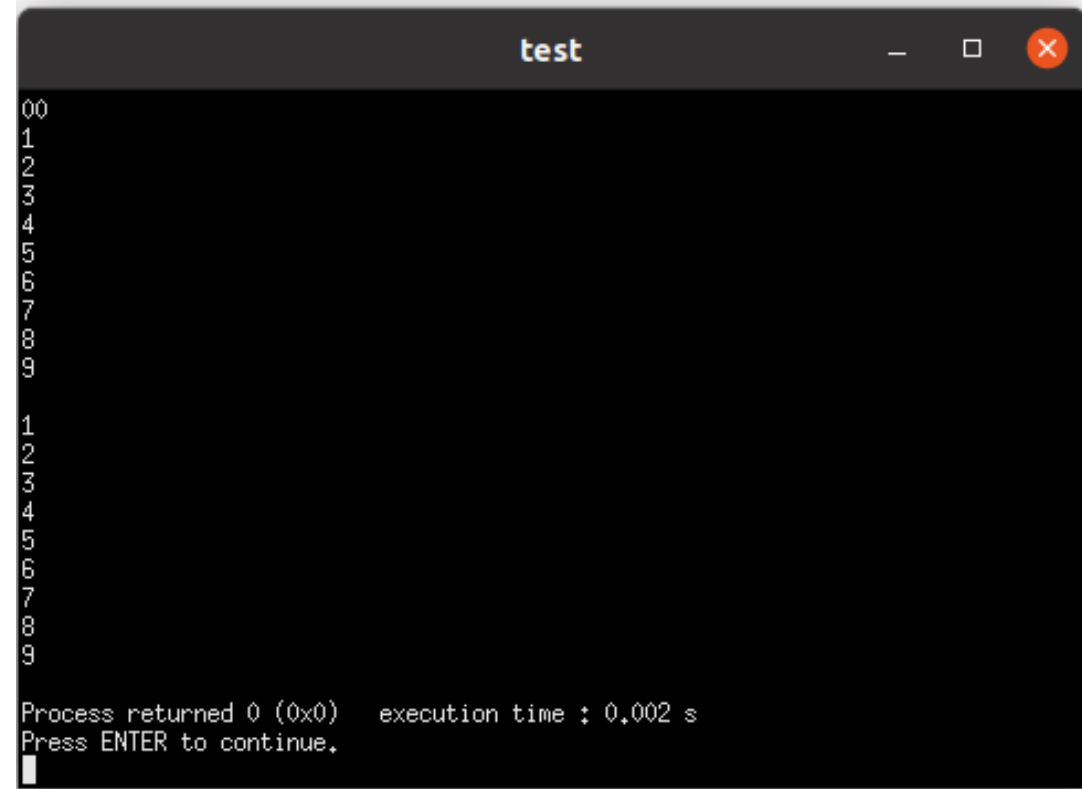
Code:Blocks test



Code:Blocks test



Code:Blocks test



```
test
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

Uvod

UDŽBENIK — STRANICE 1-9

Zadatak operativnog sistema

- Operativni sistem:

- Objedinjuje raznorodne delove računara tako što upravlja procesorom, kontrolerima i RAM-om.
- Skriva od korisnika detalje funkcionisanja tako što pretvara računar od mašine koja rukuje bitima, bajtima i blokovima u mašinu koja rukuje datotekama i procesima.

Pojam datoteke

- Datoteka ima:
 - Sadržaj (korisničke podatke)
 - Atribute (npr. veličina ili vreme kreiranja) – u deskriptoru datoteke
- Uloga datoteke:
 - Trajno čuvanje podataka.
 - Pristup podacima je čitanje i pisanje (kojima predstoji otvaranje i nakon kojih sledi zatvaranje datoteke).
- Što datoteke zatvaramo?

Pojam procesa

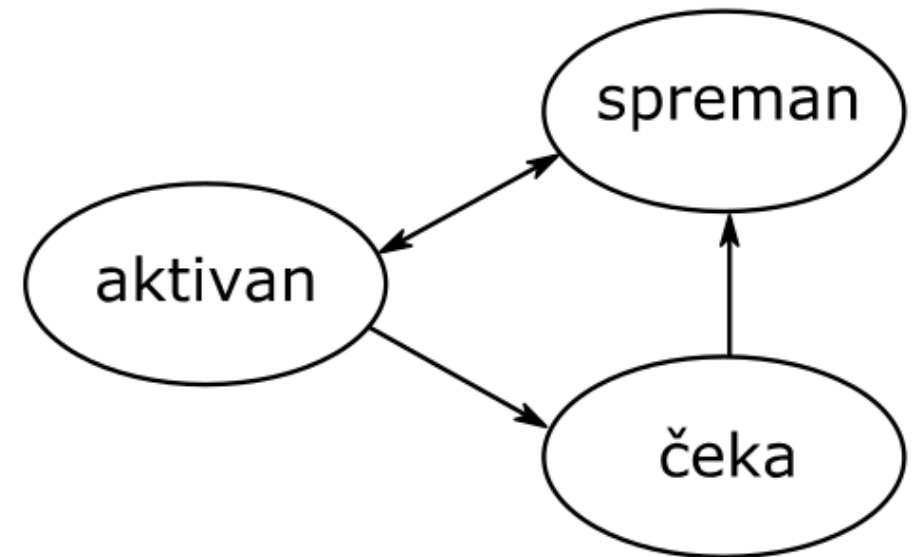
- Aktivnost procesa – angažovanje procesora na izvršavanju korisničkog programa.
- Slika procesa – adresni prostor procesa (naredbe, stek i podaci).
- Atributi – stanje, prioritet – čuvaju se u deskriptoru procesa.

Stanje i prioritet procesa 1/2

- Tipična stanja procesa su:
 - aktivan
 - čeka
 - spreman.
- Prioritet procesa određuje kada je proces aktivan:
- Uvek je aktivan proces sa najvišim prioritetom.
- Ako postoji nekoliko procesa sa najvišim prioritetom, vrši se raspodela procesorskog vremena između njih uz pomoć mehanizma kvantuma.
- Isticanje kvantuma regulišu prekidi sata.

Stanje i prioritet procesa 2/2

- Aktivan proces prelazi u stanje “čeka” kada je nemoguć nastavak njegove aktivnosti (npr. UI radnja). Nakon tog čekanja prelazi u stanje “spreman”.



Stanje i prioritet procesa, praktična ilustracija

```
top - 00:50:33 up 3 min, 1 user, load average: 0,54, 0,46, 0,20
Tasks: 199 total, 1 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,3 us, 0,3 sy, 0,0 ni, 99,4 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 16040,3 total, 13192,8 free, 285,7 used, 2561,8 buff/cache
MiB Swap: 947,2 total, 947,2 free, 0,0 used. 15445,7 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1984	veljko	20	0	572912	78812	34644	S	2,0	0,5	0:02.10	Xorg
3052	veljko	20	0	599928	38388	28120	S	2,0	0,2	0:01.47	gnome-terminal-
16863	veljko	20	0	37616	4028	3300	R	0,7	0,0	0:00.17	top
1	root	20	0	195116	9444	6660	S	0,3	0,1	0:06.71	systemd
2119	veljko	20	0	118468	2204	1824	S	0,3	0,0	0:00.15	VBoxClient
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp

Uloga procesa

- Procesi omogućuju bolje iskorišćenje računara (procesora) i njegovu bržu reakciju na dešavanje spoljnih događaja (npr. unos teksta).
- Istovremeno postojanje više procesa omogućuje da se procesor preključi sa aktivnog procesa na spreman proces kada aktivan proces prelazi u stanje “čeka”.
- Dobar primer ovakvog ponašanja je čekanje hitnog procesa na spoljni događaj (npr. unošenje teksta sa tastature).

Pojam niti

- Redosled naredbi programa (procesa) naziva se **trag (trace)** procesa.
- Proces je **sekvencijalan** ako je njegov trag poznat u **vreme programiranja**.
- Trag sekvencijalnog procesa naziva se **nit (thread)** koja **povezuje** izvršavane naredbe u **redosledu** njihovog izvršavanja.

Mana sekvencijalnih procesa

- Mana sekvencijalnih procesa je da su **neosetljivi** na **spoljne** događaje (npr. editovanje teksta).
- Za editovanje su potrebne **dve** radnje (**interakcija** sa korisnikom i **čuvanje** unešenog teksta).
- Sekvencijalan editorski proces izvršava te dve radnje jednu za drugom:

...

```
for(;;) {  
    do_editor_command();  
    if(time_to_save_data())  
        save_data();  
}
```

}

...

Nesekvencijalan editorski proces (odvojene radnje)

...

```
for(;;) {  
    do_editor_command_in_foreground();  
}
```

...

```
for(;;) {  
    if(time_to_save_data())  
        save_data_in_background();  
}
```

...

Nesekvencijalan editorski proces (odvojene radnje)

- **Prioritetnija** radnja je posvećena **interakciji** sa korisnikom, a **manje prioritetna** pozadinska radnja je posvećena **čuvanju teksta**.
- Pod pretpostavkom da je **hitna** radnja zaustavljena, jer nema komandi od korisnika, **pozadinska** radnja može da se odvija sve dok, na primer, spoljni događaj poput **pritiska dirke** na tastaturi ne najavi početak interakcije sa korisnikom.

Nesekvencijalan editorski proces (odvojene radnje)

- Tada se **zaustavlja pozadinska** radnja, radi nastavljanja **hitne** radnje.
- Kada se obavi korisnička komanda u okviru hitne radnje, a **hitna radnja se zaustavi** u očekivanju nove komande, **nastavlja se pozadinska radnja**.
- Zahvaljujući **preplitanju** hitne i pozadinske radnje, u toku editiranja **nema perioda bez odziva**.
- Podrazumeva se da opisanom nesekvencijalnom editorskom procesu odgovaraju **dve niti**.

Nesekvencijalan editorski proces (odvojene radnje)

- Da bi opisano rukovanje nitima bilo moguće, **prioritet**, **stanje** i **stek** se ne vezuju za **proces**, nego za njegove **niti**.
- Znači svaka nit procesa ima svoj **prioritet**, svoje **stanje**, svoj **stek**, pa i svoj **deskriptor**.
- Za niti istog procesa se podrazumeva da **nisu potpuno nezavisne**, odnosno da **sarađuju razmenom podataka**.
- Tako, u slučaju **nesekvencijalnog** editorskog procesa, manje prioritetna nit se brine o **čuvanju** teksta koga **pripremi** prioritetna nit.

Konkurentni procesi

- Procesi sa **više niti** nazivaju se **konkurentni procesi (konkurentni programi)**.
- Podrazumeva se da samo jedna od niti može biti “**aktivna**”, dok su ostale u stanju “**spremna**” ili “**čeka**”.
- Preključivanje procesora sa jedne niti na drugu, uzrokuju **redosled izvršavanja koji nije određen u vreme programiranja**, što znači da je izvršavanje konkurentnih procesa u opštem slučaju **stohastično** zbog stohastične prirode dešavanja spoljnih događaja.

Struktura operativnog sistema

- Zadatak operativnog sistema je da upravlja fizičkim i logičkim delovima računara u okviru svog **jezgra (kernela)**.
- **Fizičkim** delovima upravljaju moduli za rukovanje:
 - Procesorom
 - Kontrolerima
 - Radnom memorijom
- **Logičkim** delovima upravljaju moduli za rukovanje:
 - Datotekama
 - Procesima

Modul za rukovanje procesorom

- Zadatak ovog modula je **preključivanje** jedne niti na drugu.
- Moguće je preključivanje na niti u **istom procesu** ili u **različitim procesima**.
- Preključivanje procesora između niti **istog procesa** je **brže** nego preključivanje niti u okviru **različitih procesa** zato što se niti **istog procesa** nalaze u **istom adresnom prostoru**.
- Ovaj modul uvodi **operaciju preključivanja**.

Modul za rukovanje kontrolerima

- Zadatak ovog modula je upravljanje **ulaznim** i **izlaznim** uređajima koji su zakačeni za kontrolere.
- Modul se sastoji od niza komponenti nazvanih **drajveri**.
- Cilj drajvera jeste da uređaje predstavi u **apstraktnom** obliku sa **jednoobraznim** i **pravilnim** načinom korišćenja (primer drajvera diska).
- Drajveri uvode operacije **ulaza** i **izlaza**, u okviru kojih se rukuje **preključivanjem (zaustavljanjem)** niti koja je zatražila operaciju.
- Drajveri takođe rukuju i **obradom prekida** kao reakcijom na javljanje uređaja putem mehanizma **prekida**.

Modul za rukovanje radnom memorijom

- Zadatak ovog modula je da vodi evidenciju o **slobodnoj** radnoj memoriji radi **zauzimanja** i **oslobađanja**.
- U slučaju da podržava **virtuelnu memoriju**, ovaj modul se brine i o prebacivanju sadržaja između **radne** i **masovne memorije**.
- Ovaj modul uvodi operacije **zauzimanja** i **oslobađanja**.

Modul za rukovanje datotekama

- Zadatak modula za rukovanje datotekama je da omogući **otvaranje** i **zatvaranje** datoteka, kao i **čitanje** i **pisanje** njihovog sadržaja.
- Ovaj modul vodi evidenciju o **blokovima (HDD)** u kojima se nalaze sadržaji datoteka.
- Takođe ovaj modul vodi računa o **prebacivanju sadržaja između radne i masovne memorije** uz pomoću operacija **čitanja (ulaza)** i **pisanja (izlaza)**, kao i **bafera** potrebnih za smeštanje sadržaja (modul za rukovanje memorijom).
- Pored ovog modul uvodi i operacije **otvaranja** i **zatvaranja**.

Modul za rukovanje procesima

- Zadatak ovog modula je da omogući **stvaranje** i **uništavanje procesa**, kao i **stvaranje** i **uništavanje** njihovih **niti**.
- Na ovaj način se uvodi **višeprocesni** i **višenitni** režim rada koji omogućava:
 - **Bolje iskorišćenje procesora**
 - **Podršku većeg broja korisnika**
 - **Bržu reakciju na spoljne događaje**
- Modul za rukovanje procesima poziva i operacije drugih modula (**za upravljanje datotekama i memorijom**).
- Modul za rukovanje procesima uvodi operacije **stvaranja** i **uništavanja (procesa i niti)**.

Slojeviti operativni sistem

- Za razliku od **slojevitog** operativnog sistema u praksi se uglavnom sreću **monolitni** operativni sistemi, koji **nemaju hijerarhijsku strukturu** jer saradnja **nije ograničena** kao kod slojevitog OS.

modul za rukovanje procesima
modul za rukovanje datotekama
modul za rukovanje radnom memorijom
modul za rukovanje kontrolerima
modul za rukovanje procesorom

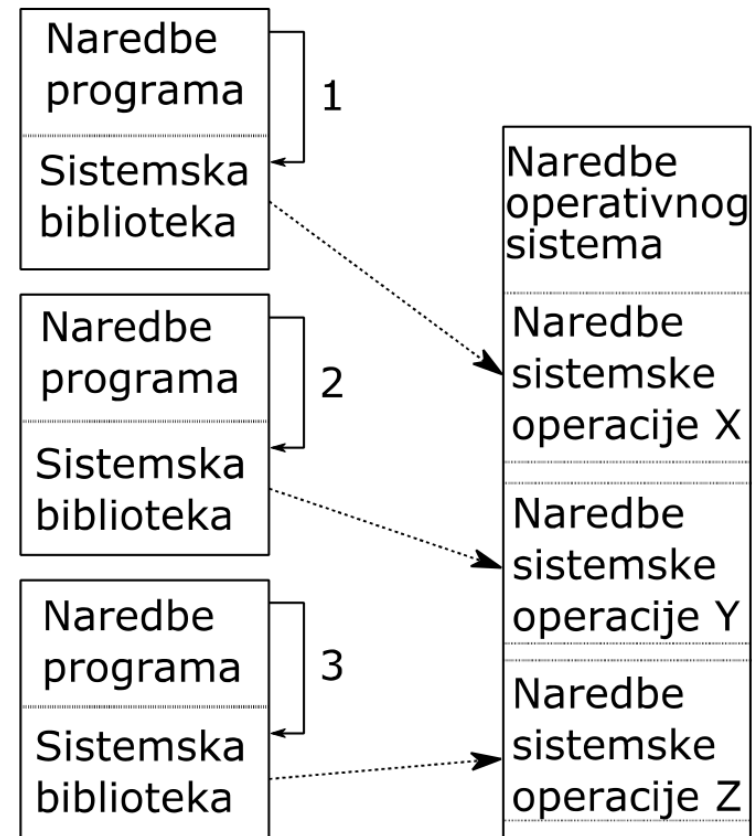
Sistemske pozivi

- Svaki proces se nalazi u **korisničkom sloju** (gornjem sloju OS), i poseduje poseban adresni prostor koji se naziva **korisnički prostor (user space)**.
- Operativni sistem poseduje poseban adresni prostor koji se naziva **sistemske prostor (kernel space)**.
- Zbog razdvojenosti ova dva prostora neophodno je uvođenje **sistemskih poziva, radi poziva operacija OS**.
- Sistemske pozivi zahtevaju korišćenje **ASM** naredbi i sakrivaju se unutar **sistemskih potprograma (sistemskih operacija)**.
- Svaki proces u sistemskom prostoru ima svoj **sistemske stek**.
- Sistemske potprogrami obrazuju **sistemske biblioteku**.

Sistemske pozivi

- Zahvaljujući **sistemskim potprogramima**, odnosno **sistemske biblioteke**, operativni sistem predstavlja **deo korisničkog programa**, iako za njega nije direktno linkovan.
- Istovremeno **postojanje više procesa** i **nepredvidivost preključivanja**, uzrokuje da je moguće da istovremeno postoji **više procesa**, koji su **započeli**, a **nisu završili** svoju aktivnost u okviru **operativnog sistema**, odnosno, čija aktivnost je **zaustavljena unutar sistemskih operacija** operativnog sistema.

Preplitanje izvršavanja tri systemske operacije



Praktična ilustracija sistemskih poziva

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

int main(){
    unsigned char buff[32];
    ssize_t count = 0;
    int fd = open("open.c", O_RDONLY);
    while(count = read(fd, buff, 31)){
        buff[count] = '\0';
        printf("%s", buff);
    }
    close(fd);
}
```

Interakcija korisnika i OS

- **Komande** komandnog jezika omogućavaju korišćenje OS na **interaktivnom** nivou.
- Za interpretiranje i preuzimanje komandi komandnog jezika zadužen je poseban proces iz korisničkog sloja koji se zove **interpreter komandnog jezika (shell)**.
- Interpreter komandnog jezika koristi OS na programskom nivou, jer u toku svog rada poziva **sistemske operacije**.