

*2022/23*

*PREDMET: CS450 KLAUD RAČUNARSTVO*

Domaći zadatak: **2**

Ime i prezime: **Dušan Stanković**

Broj indeksa: **3611**

Datum izrade: **14.05.2023.**

# Tekst domaćeg zadatka:

1. Preuzeti, instalirati i podesiti Docker;
2. Opisati i propratiti screenshot–ovima u Word dokumentu navedene procese;
3. Kreirati direktorijum sa izvornim kodom java aplikacije i pokrenuti ga pomoću Dokera. Treba da ispiše tekst "CS450 - Klaud Kompjuting";

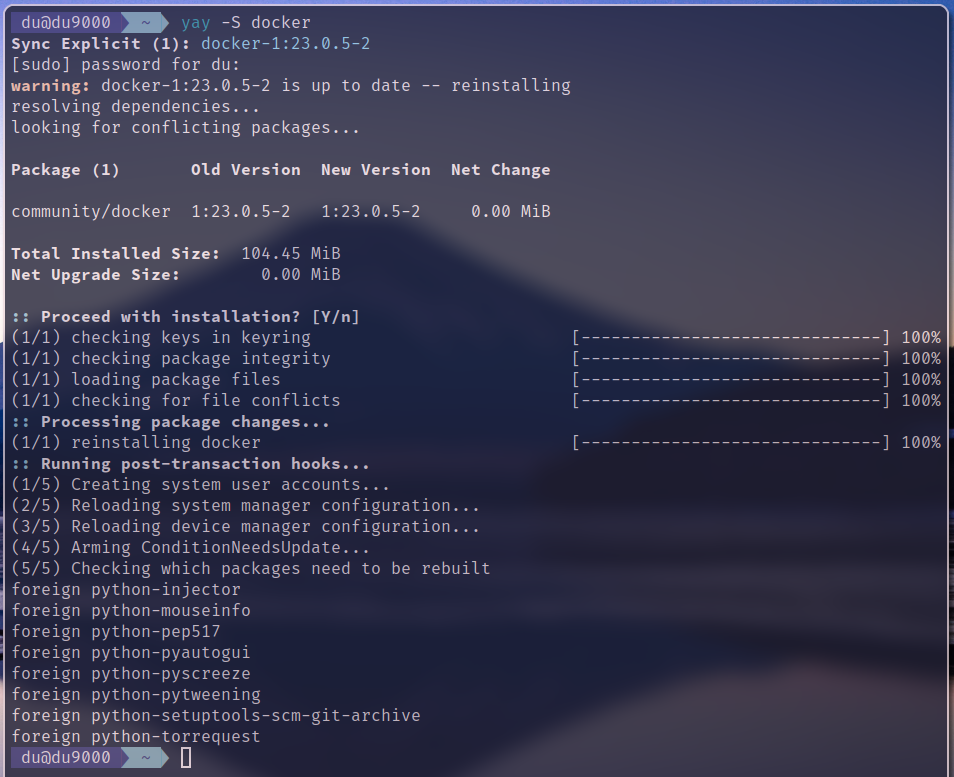
4. Opisati i propratiti screenshot–ovima u prethodno kreiranom Word dokumentu (iz tačke 2) sve navedene procese (iz tačke 3);

# Rešenje zadatka:

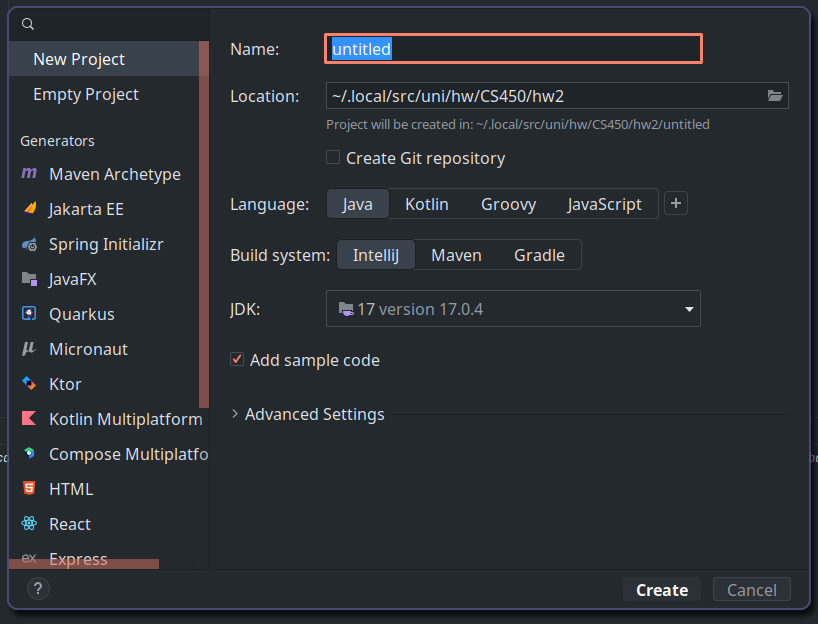


Za izradu ovog domaćeg zadatka koristim Arch based Linux distribuciju koja koristi yay package manager, koji je u suštini samo wrapper pacman package managera.

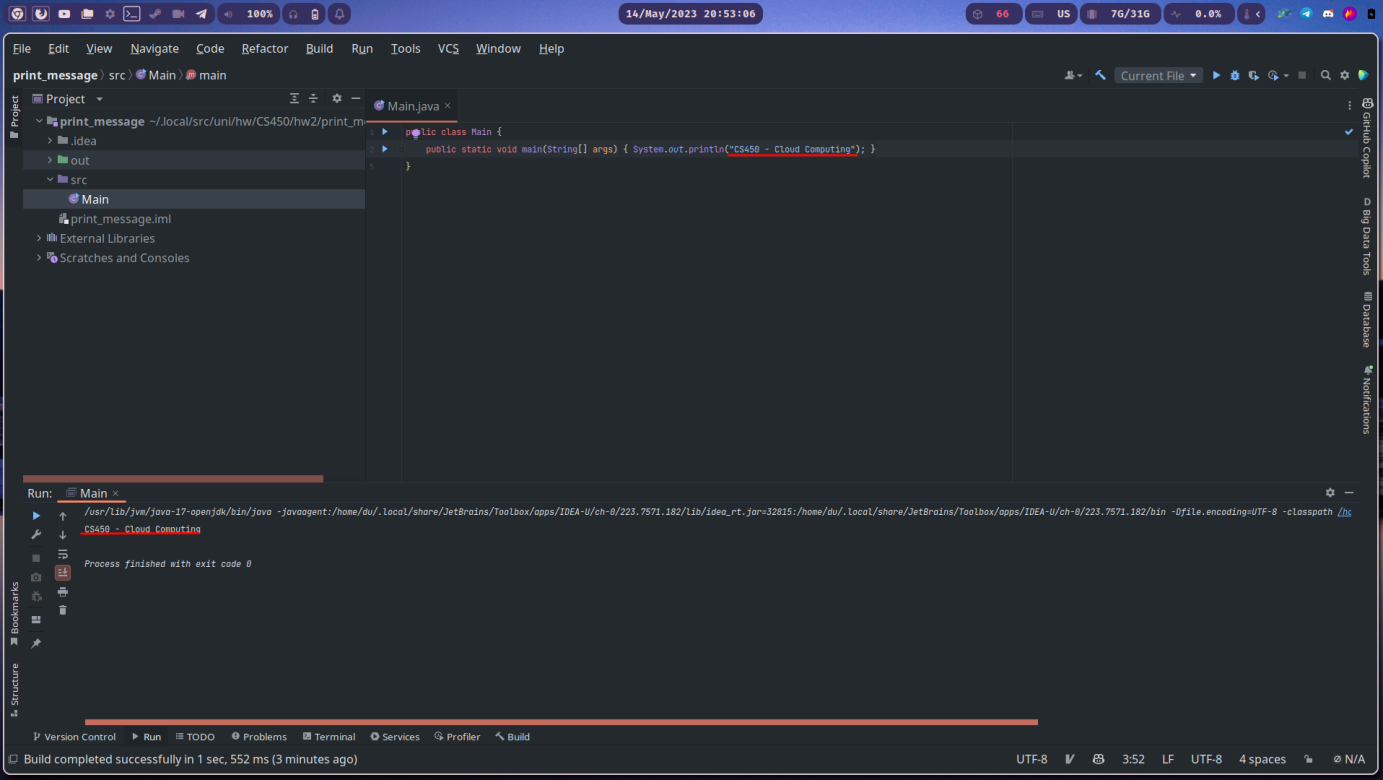
Na prvoj slici vidimo rezultate pretrage za paketom pod nazivom “docker” izvršeno komandom yay -Ss docker, i vidimo da u community repozitorijumu postoji docker paket koji je već instaliran po defaultu na svim arch based distribucijama.



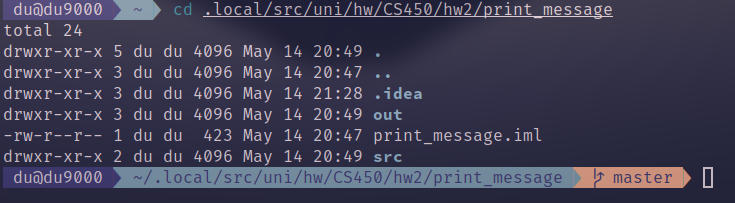
Ovako izgleda proces instalacije, Net upgrade size je 0 Mibibajta jer sam već imao najnoviju verziju instaliranu, ali komanda yay -S docker koja služi za instaliranje paketa, takođe može služiti i za apdejtovanje ako taj paket već imamo kao u ovom slučaju.



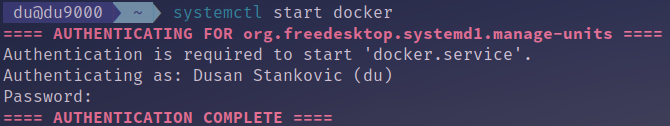
Kreiramo novi java projekat koristeći JDK 17 verziju jave.



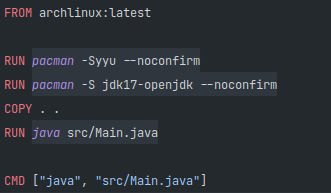
Testiramo jednostavnu java aplikaciju kako bi lokalno videli output koji želimo da vidimo i pri pokretanju docker java image-a.



docker_6Korišćenjem CLI prelazimo u direktorijum projekta i kreiramo Dockerfile.

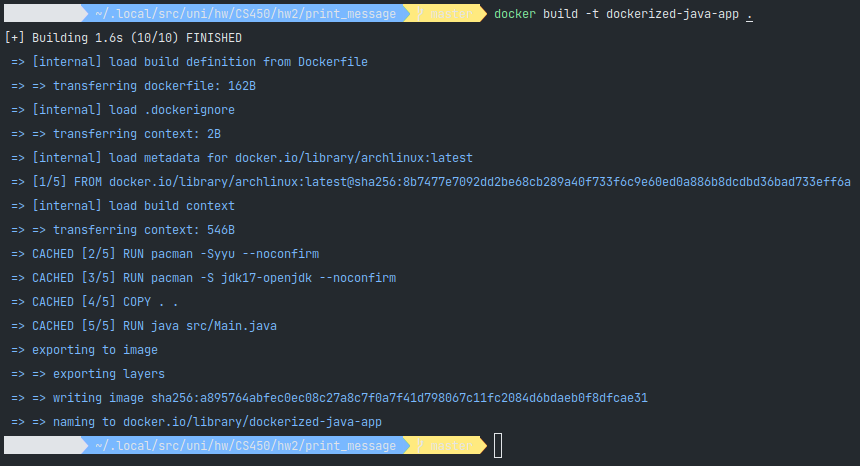


Pokrećemo docker daemon servis kako bi mogli da kreiramo i pokrenemo image.

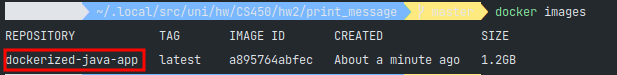


Ovako izgleda naš dockerfile, u ovom slučaju environment koji koristim je archlinux, prva komanda koju pokrećemo je pacman -Syyu, moguće je koristiti i pacman -Syu koji samo apdejtuje sve pakete, ali je sigurnije koristiti -Syyu koji apdejtuje i sve repozitorijume paketa kao što su aur, community, extra itd. Nakon toga instaliramo jdk verziju koju smo lokalno koristili za pokretanje java aplikacije, iako je jednostavna i bilo koja druga bi radila. Noconfirm flag koristimo kako prompt ne bi čekao potvrdu, već će prihvatiti default odabranu opciju, za instalaciju što je u ovom slučaju Y. Takođe ne moramo brinuti o postavljanju globalnog i home path-a za javu, jer se na arch linux distribucijama setuje kao deo instalacije paketa.

Na kraju postavljamo parametre java i path do samog fajla koji se koriste tokom pokretanja kontejnera.



koristimo komandu docker build sa flagom -t što znači tag i imenujemo ga dockerized-java-app i argumentom “.” odabiramo sve fajlove iz direktorijuma.



Proveravamo da li se ispravno kreirao image korišćenjem komande docker images, koji nam izlista sve image.

docker_11

I konačno komandom docker run dockerized-java-app dobijamo željeni output kao kada smo lokalno testirali aplikaciju.