Operativni Sistemi

II Semestar - 2022/23 - Vježbe

Sedmica 1

Handout za Vježbe

Agenda:

- PAPP
- Turingova Mašina
- Arhitektura Osnovih Operativnih Sistema
- Sistemi za verzioniranje koda (VCS) – GIT
- Manuelna Kompilacija

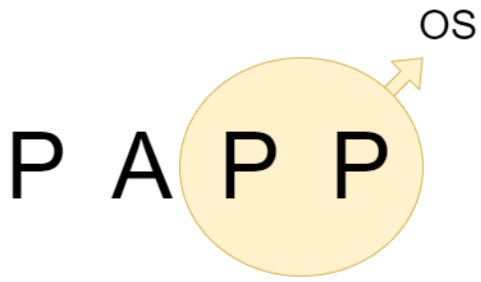
Kontakt:

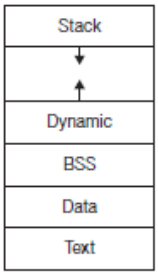
Narcisa.hadzajlic@size.ba (A Grupa)

Adin.jahic2019@size.ba (B Grupa)

Problem – Algoritam – Program – Proces (PAPP)

PAPP model predstavlja vizuelnu reprezentaciju životnog ciklusa i misaonog procesa koji je potreban za razumijevanje abstraktnih pojmova u svim poljima informacijskih tehnologija. Sve oblasti koje se nalaze u oblasti Informacijskih tehnologija se mogu simulirati preko ovoga akronima. Oblast operativnih sistema obuhvata ponašanje **programa** i njihovu transformaciju u **procese** te njihovu sinhronizaciju i upotrebu sa stajališta inžinjera.





Kroz rad sa Programiranjem smo upoznati sa arhitekturom Programa, Nekakva osnovna vizuelna reprezentacija procesa koji možemo zamisliti u glavi se svodi trivializovan diagram poput ovoga:

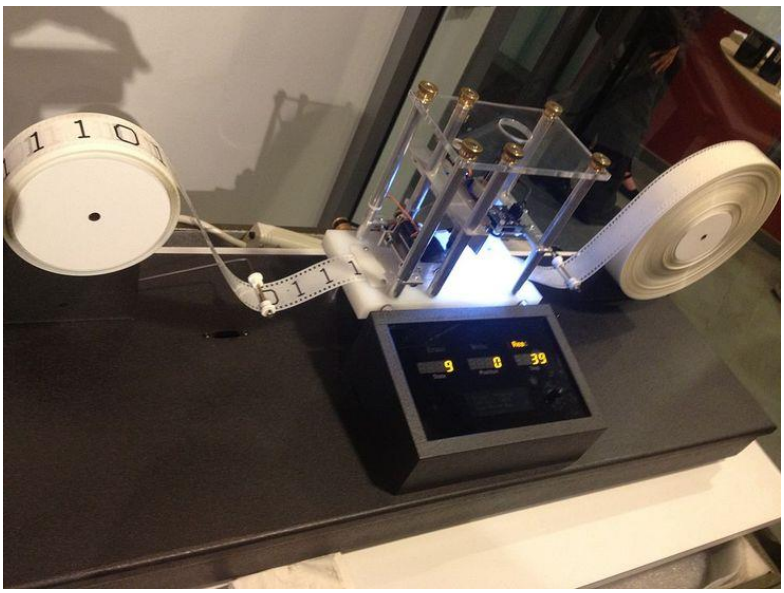
Turingova Mašina

Razumijevanje operativnih sistema se zasniva na razumijevanju interne strukture rada računara. Jedan od najboljih primjera za opisivanje rada računara je preko **Turingove Mašine** koja predstavlja abstraktni koncept generalnog modela računarske komputacije. Trivializovan pristup sa stajališta Operativnih sistema opisuje ovu mašinu kroz nekoliko osnovnih komponenata:

- Traka sa simbolima
- Instrukcijski set
- Glava koja vrši analizu

Traka je podijeljena u segmente na kojima se nalaze **simboli,** Glava se pozicionira iznad segmenta trake, registruje simbol, pretražuje simbol u Instrukcijskom setu, provjerava stanje mašine (stanja mašine su zadana u instrukcijskom setu), te glava djeluje po instrukcijama vezanim za simbol i stanje mašine zadanim u instrukcijskom setu (pomjeranje lijevo ili desno na traci).

Ovaj model komputacije predstavlja generalni prikaza rada CPU i po logici izvršavanja nije ništa drugačiji od najmodernijih procesora na današnjem tržištu.

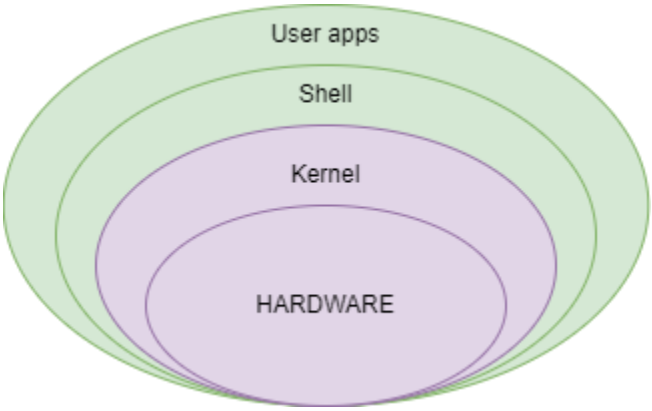


Arhitektura Osnovnih Operativnih Sistema (LINUX)

Močna turingova mašina daje vizuelni prikaz defincije pojmova programa i procesa.Nekakva osnovna definicija procesa jeste da je to „program koji se pokrene ili učita u obradivu memoriju“. Program predstavlja naš kod koji ispišemo u bilo kojem programskom jeziku. Aktivacija njegova je ono što nam daje mogucnost da ga oslovljavamo kao **Proces**. Operativni sistemi generalno imaju 2 moda rada,a to su: **Korisniči režim** i **privilegovani režim** (User mode and Privilegued mode). Najlakša distrinkcija je u tome što Korisnički dio nema pristup cijelom operativnom sistemu, dok privilegovani.Nekakav tradicionalni način konceptualiziranja ovih koncepata je da svi programi koji pripadaju operativnom sistemu i koji su sklopu operativnog sistema te po ekstenziji CIJELI operativni sistem sam po sebi bez korisnickih dodataka pripada Privliegovanom režimu rada, dok sve korisničke aplkacije se su u korisničkom režimu rada.

Poznata riječ „KERNEL“ se odnosi na segment operativnog sistema koji je u **privilegovanom režimu.**

Makroskopska vizuelna apstrakcija arhitekture operativnog sistema se može prikazati na sljedeći način:



Zelenom bojom je označen segmenat operativnog sistema koji je u **Korisničkom Režimu**, dok je Ljubičastom bojom označen segmenat operativnog sistema koji je u **Privilegovanom Režimu.**

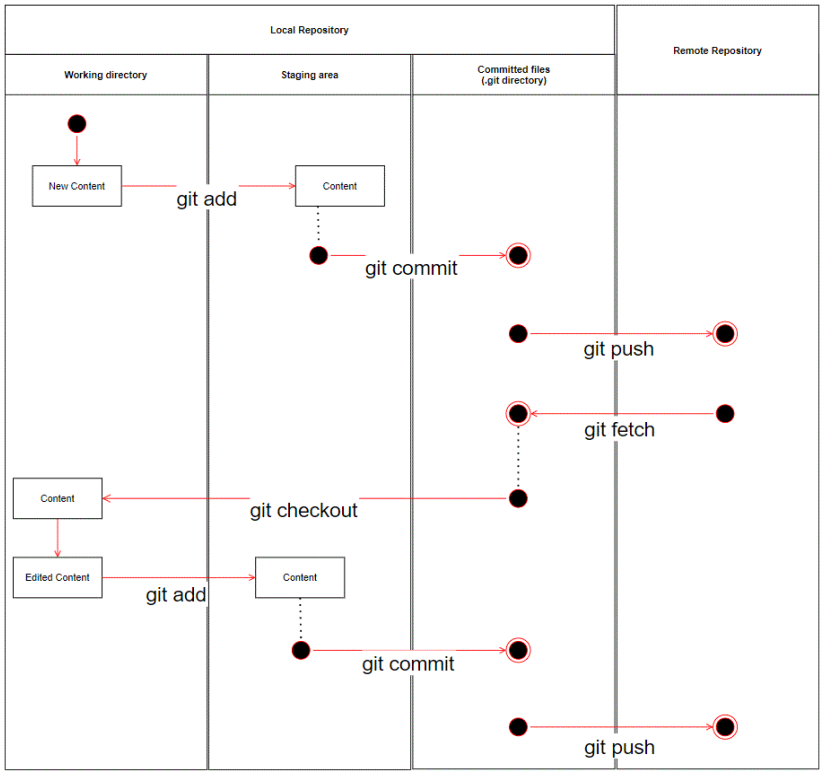
Iz ovoga na slici može se intutivno zaključiti definicija Kernela kao **INTERFEJS** između korisničkog režima rada i hardware-a računara. On se brine o drajverima, sistemskim pozivima dubokim arhitekturama operativnih sistema koje bi nama kao Sistemskim inžinjerima bile previše komplekse za svakodnevnu upotrebu.

Sistemi za Verzioniranje koda (VCS)

Najbolji predstavnik VCS Sistema u današnjem vremenu je **GIT**. On omogučava stvari poput:

- Sinhronizacije koda i napretka na projektima
- Praćenje verzija programa
- Protokol u slučaju nezgode (Disaster recovery)
- Može u ekstremnim situacijama zamijeniti sisteme za autorizaciju, te poslužiti kao RBAC(Role Based Access Control)

Zasnovan je na principu rada sa repozitorijumima. Trivializovano . Repozitorij je jedinica mjere GIT-a koji služi kao spremnik za naš kod (analogno projektnim direktorijumima koje inače koristimo).Komunikacija sa gitom se izvodi radom preko **shella** i koristi instacu shella pod nazivom **BASH (**Bourne Again Shell). Svaka komanda unutar Shella pocinje sa Git <command>. Spašavanje progresa se radi pomoču sisteme pod nazivom „COMMIT“, sve promjene koje se „komitaju“ budu spašene i dobiju jedinstveni short SHA256 kod kao jedinstveni identifikator koji nam može sluziti za vračanje koda. Svaki „komit“ zahtjeva poruku kao verbalni opis koji ide uz promjenu. Vizuelna reprezentacija izgleda ovako:



Remote u zadanom dijagramu predstavlja repozitorij napravljen na internetu (CLOUD) na platformi koja vrši skladišetenje repozitorija poput GITHUB, BITBUCKET,GITLAB... Sinhronizacija između remote-a i localnog repozitorija se vrši pomoču 2 komande GIT PUSH I GIT PULL koji dobavljaju posljednje komitove sa REMOTE-a i analogno šalju posljednji komit na REMOTE sa lokalnog repozitorija.

Manuelna Kompilacija

Da biste manuelno kompajlirali C++ kod, slijedite ove korake:

Otvorite uređivač teksta i napišite svoj C++ kod. Spremite ga sa ekstenzijom .cpp, na primjer, "myprogram.cpp".

Otvorite terminal i navigirajte do direktorija u kojem se nalazi vaš kod.

Kompajlirajte svoj kod koristeći C++ kompajler, kao što je g++. Na primjer, ako imate g++ kompajler instaliran na svom sistemu, možete kompajlirati svoj kod sljedećom naredbom:

- g++ myprogram.cpp -o myprogram

Ova naredba će kompajlirati vaš kod i stvoriti izvršnu datoteku nazvanu "myprogram" u istom direktoriju.

Ako u vašem kodu postoje greške, kompajler će ih prikazati u terminalu. Morat ćete popraviti greške prije nego što uspješno kompajlirate svoj kod.

Nakon što je vaš kod uspješno kompajliran, možete ga pokrenuti tako da u terminal upišete njegovo ime, a zatim sve potrebne argumente. Na primjer, za pokretanje "myprogram" upisali biste:

- ./myprogram

Ovo će izvršiti vaš program i prikazati bilo kakav izlaz u terminalu.

Napomena: Tačne naredbe i sintaksa za kompajliranje C++ koda mogu varirati ovisno o vašem operativnom sistemu i kompajleru koji koristite.

Za sve eventualne primjedbe, komentare, sugestije obratiti se na mail: narcisa.hadzajlic@dl.unze.ba