Sigurnost u računarskim mrežama

Modeli sigurnosti

Pojam i problem bezbednosti i modeli sigurnosti

Nemanja Maček

Sadržaj

- Pojam i problem bezbednosti
- Značajniji pojmovi
- Modeli kontrole pristupa
- Modeli integriteta
- Dodatak: neformalni opis Tjuringove mašine

Razlika između bezbednosti i sigurnosti

- Pitanja:
  - Kako možemo da odredimo da li je računarski sistem siguran ili nije?
  - Da li postoji generički aloritam koji omogućava da odredimo da li je računarski sistem siguran?
  - Šta podrazumevamo pod pojmom „siguran“?
- Razlika između pojmova bezbednost i sigurnost sastoji se u sledećem:
  - **Bezbednost** (engl. *safety*) se odnosi se na apstraktni model.
  - **Sigurnost** (engl. *security*) se odnosi na aktuelnu implementaciju.
- Siguran sistem odgovara modelu koji je bezbedan u odnosu na sva prava.
- Međutim, model bezbedan u odnosu na sva prava ne garantuje siguran sistem.

Razlika između sigurnosti i privatnosti

* Slika preuzeta sa foruma: http://forum.dvdtalk.com/

Pojam bezbednosti

- Iskoristićemo matricu kontrole pristupa da definišemo pojam bezbednosti.
- Neka je *R* skup generičkih (prostih) prava pristupa na sistemu, bez specijalnih prava kopiranja objekata i vlasništva nad objektima.
- Ukoliko dodavanje generičkog prava $r \in R$ elementu matrice za kontrolu pristupa može da stvori sugurnosni propust u sistemu, onda se kaže da to pravo **kompromituje (ugrožava) sistem**.
  - Ako pravo *r* nikada, ni na koji način, ne može da ugrozi sistem, onda se sistem smatra **bezbednim** u odnosu na to pravo.
  - U suprotnom, sistem se ne smatra bezbednim u odnosu na to pravo.

Problem bezbednosti

- **Problem bezbednosti** može se definisati pomoću sledećeg pitanja:
  - Postoji li algoritam pomoću koga ćemo odrediti da li je posmatrani sistem zaštite sa inicijalnim stanjem s_0 bezbedan u odnosu na generičko pravo *r*?
- Odgovor na ovo pitanje daju sledeće dve teoreme.
- **Teorema 1**.
  - Postoji algoritam pomoću koga se može odrediti da li je dati monooperativni sistem (sistem **ograničen na neki način**) sa inicijalnim stanjem s_0 bezbedan u odnosu na pravo *r*.
  - Dokaz:
    - Svaka komanda je identifikovana primitivnom operacijom koju proizvodi.
    - Pretpostavite minimalnu sekvencu komandi, neophodnu da pravo *r* ugrozi ovakav sistem koji se nalazi u početnom stanju s_0 .
    - Može se dokazati da je dužina ove sekvence konačna.
    - To znači da je moguće odrediti i sva stanja u kojima se sistem može naći i da se može odrediti da li je sistem bezbedan ili ne.

Problem bezbednosti

- **Teorema 2**.
  - Ne može se odrediti da li je **generički** sistem zaštite bezbedan za dato generičko pravo.
  - Dokaz:
    - Pretpostavite da Tjuringovu mašinu možemo da svedemo na problem bezbednosti, tako da konačno stanje mašine odgovara kompromitovanju sistema generičkim pravom *r*.
    - Ukoliko je problem sigurnosti rešiv, može se odrediti kada će se Tjuringova mašina zaustaviti.
    - Međutim, pošto se u ovom slučaju govori o funkciji koja nije Tjuring-izračunljiva, problem bezbednosti sistema je nerešiv.
- Dakle, problem bezbednosti je:
  - **Neodređen** za **generičke** modele zaštite,
  - **određen** za modele koji su **ograničeni** na neki drugi način.
- Pitanje: da li se sećate šta je Tjuringova mašina? Ukoliko se ne sećate, pogledajte dodatak ovim beleškama.

Značajniji pojmovi

- **Sigurnosna arhitektura** informacionog sistema je osnova za sprovođenje sigurnosne politike svake organizacije.
  - Zavisno od toga o koliko strogoj sigurnosnoj politici je reč i o kakvom se sistemu radi gradi se odgovarajuća sigurnosna arhitektura i primenjuju se adekvatne metode zaštite.
- **Otvoren sistem** ne zavisi od proizvođača.
  - Specifikacije su objavljene, što omogućava saradnju sa proizvodima drugih proizvođača.
  - Prednost otvorenog sistema u pogledu sigurnosti jeste to što je podložan javnom pregledu, tj. nezavisnom ocenjivanju koje najčešće pomaže u otkrivanju ranjivosti tog proizvoda.
- **Zatvoren sistem** koristi softver i harvder koji zavisi od proizvođača, i koji obično nije kompatibilan (udruživ) s drugim sistemima i komponentama.
  - Deo sigurnosti zatvorenih sistema zasniva se na uverenju da će nepoznavanje detalja arhitekture i implementacije povećati otpornost sistema na napade.
  - To se može i drugačije protumačiti: zatvoreni sistemi nisu podložni nezavisnom ocenjivanju, što znači da mogu biti ranjivi i imati slabe tačke koje trenutno nisu poznate ili otkrivene, ali se u budućnosti mogu eksploatisati.

Značajniji pojmovi

- **Računarska baza od poverenja** (engl. *Trusted Computing Base*, TCB) je kombinacija zaštitnih mehanizama unutar računarskog sistema, koja uključuje hardver, softver i firmver, i za koju se veruje da obezbeđuje primenu sigurnosnih pravila.
- **Sigurnosni perimetar** (engl. *security perimeter*) je granica koja odvaja TCB od ostatka sistema.
- **Nerizičan put** (engl. *trusted path*) obezbeđuje korisniku da pristupi TCB-u tako da ga pri tome ne mogu kompromitovati drugi procesi i/ili korisnici.
- **Računarski sistem od poverenja** (engl. *trusted computer system*) je računarski sistem koji koristi nužne mere obezbeđenja hardvera i softvera kako bi omogućio obradu informacija klasifikovanih na više nivoa.
  - Ovaj sistem treba da zadovolji specificirane zahteve u pogledu pouzdanosti i sigurnosti.
- **Sigurnosna oznaka** (engl. *security label*) se dodeljuje nekom resursu.
  - Može ukazati na potrebu za posebnim načinom (odnosno režimom) rukovanja (rukovanje objektom uz primenu dodatnih sigurnosnih mehanizama).
  - Može se koristiti za kontrolu pristupa.

Modeli sigurnosti informacija

- Kao način da se formalizuju sigurnosna pravila, često se koriste modeli.
- Ovi modeli mogu biti apstraktni ili intuitivni i obezbeđuju okvir za razumevanje osnovnih koncepata.
- Modeli, od kojih će neki ukratko biti opisani u nastavku izlaganja, se mogu podeliti na:
  - **Modele kontrole pristupa** (engl. *access control models*)
    - Bell-LaPadula model
    - Model matrice pristupa (engl. *access matrix*)
    - Model preuzmi-dodeli (engl. *take-grant model*)
  - **Modele integriteta** (engl. *integrity models*)
    - Biba model integriteta
    - Clark-Wilson model integriteta
  - **Modele toka informacija** (engl. *information flow models*)
    - Model bez preplitanja (engl. *non-interference model*)
    - Teorije kompozicije (engl. *composition theories*)

Bell-LaPadula model

- Bell-LaPadula (BLP) model se fokusira na poverljivost klasifikovanih informacija, za razliku od Biba modela integriteta, koji opusije pravila za zaštitu integriteta informacija.
- BLP opisuje skup prava za kontrolu pristupa korišćenjem **sigurnosnih oznaka** nad objektima, od najosetljivijih u pogledu tajnosti, do onih najmanje osetljivih, sa sledećom kategorizacijom:
  - strogo poverljivo (engl. *top secret*),
  - tajna (engl. *secret*),
  - poverljivo (engl. *confidential*),
  - neklasifikovano (engl. *unclassified*).
- BLP model izgrađen je na konceptu **konačnog automata** sa skupom raspoloživih stanja u sistemu, pri čemu su tranzicije iz stanja u stanje definisane funkcijama tranzicije.
- U ovom formalnom modelu:
  - Entiteti informacionog sistema podeljeni su na **subjekte** i **objekte**.
  - Definisan je pojam **sigurnog stanja**.
  - Dokazano je da svaka **promena stanja** (tranzicija iz stanja u stanje) čuva sigurnost kretanjem iz sigurnog stanja u novo sigurno stanje, time induktivno dokazujući da je sistem siguran.

Bell-LaPadula model

- Stanje sistema je definisano kao **sigurno** ako su dozvoljeni načini pristupa subjekata objektima u skladu sa sigurnosnom politikom tj. pravilima.
- Da bi se odredio dozvoljeni način pristupa, dozvole koje ima subjekat upoređuju se sa klasifikacijom objekata kako bi se odredilo da li je subjekat ovlašćen za određeni način pristupa.
- Model definiše dva obavezna pravila za kontrolu pristupa i jedno diskreciono pravilo kontrole pristupa sa tri sigurnosna svojstva:
  - **Jednostavno svojstvo sigurnosti** (engl. *simple security property*). **Nema čitanja prema gore** (*no read-up*). Subjekat određenog nivoa poverljivosti ne može čitati objekat koji je na višem nivou poverljivosti.
  - **Zvezdica (*) svojstvo sigurnosti** (engl. *star security property*). **Nema pisanja prema dole** (*no write-down*). Subjekat određenog nivoa poverljivosti ne može pisati ni u jedan objekat na nižem nivou poverljivosti.
  - **Diskreciono svojstvo sigurnosti** (engl. *discretionary security property*) koristi matricu pristupa da specificira diskreciona prava.

Bell-LaPadula model

- Prenos informacija od niže osetljivosti do više osetljivosti u Bell-LaPadula modelu može se ostvariti preko koncepta **poverljivih subjekata** (*trusted subjects*).
  - Poverljivi subjekat može povrediti * svojstvo ako namena polise nije povređena.
- Ovaj sigirnosni model okarakterisan je frazom: nema čitanja prema gore i nema pisanja prema dole.
- Drugim rečima, prema BLP modelu:
  - Korisnici mogu da prave sadržaj samo na svom sigurnosnom nivou ili iznad njega.
    - Primer: subjekat nivoa tajna može praviti objekte nivoa tajna ili strogo poverljivo, ali ne i objekte nivoa poverljivo ili neklasifikovano.
  - Korisnici mogu videti samo sadržaj svog sigurnosnog nivo ili ispod njega.
    - Primer: subjekat nivoa tajna može pročitati objekte nivoa tajna, poverljivo ili neklasifikovano, ali ne može pročitati objekte nivoa strogo poverljivo.

Bell-LaPadula model

- Nedostaci BLP modela su sledeći:
  - Model razmatra normalne kanale za razmenu informacija, ali ne i skrivene, tj. tajne kanale.
  - Model ne specificira kako treba da se radi sa deljenim datotekama i serverima u modernim distribuiranim sistemima.
  - Model ne definiše eksplicitno šta je sigurna tranzicija iz jednog stanja u drugo (engl. *secure state transition*).
  - Model se zasniva na sigurnosnoj politici sa više nivoa i ne razmatra druge sigurnosne politike koje neka organizacija može zahtevati.

Model matrice pristupa

- **Matrica pristupa** na jednostavan način određuje subjektima prava pristupa i korišćenja objekata (na primer, pravo čitanja, upisa ili izvršavanja).
  - **Subjekat** je aktivni entitet (na primer, korisnik, program ili proces koji traži prava za pristup i korišćenje resursa ili objekta).
  - **Objekat** je pasivni entitet (na primer, datoteka ili neki resurs za smeštanje podataka.)
  - U nekim slučajevima, jedan entitet tj. stavka može biti subjekat u jednom kontekstu a objekat u drugom kontekstu.

subjekat / objekat	datoteka **prihod**	datoteka **plate**	proces **račun**	štampač **lab 404**
korisnik **Mile**	čitanje	čitanje / upis	izvršavanje	štampanje
korisnik **Lale**	čitanje / upis	čitanje	nikakva prava	štampanje
proces **provera**	čitanje	čitanje	izvršavanje	nikakva prava
program **obračun**	čitanje / upis	čitanje / upis	pozivanje	štampanje

Model matrice pristupa

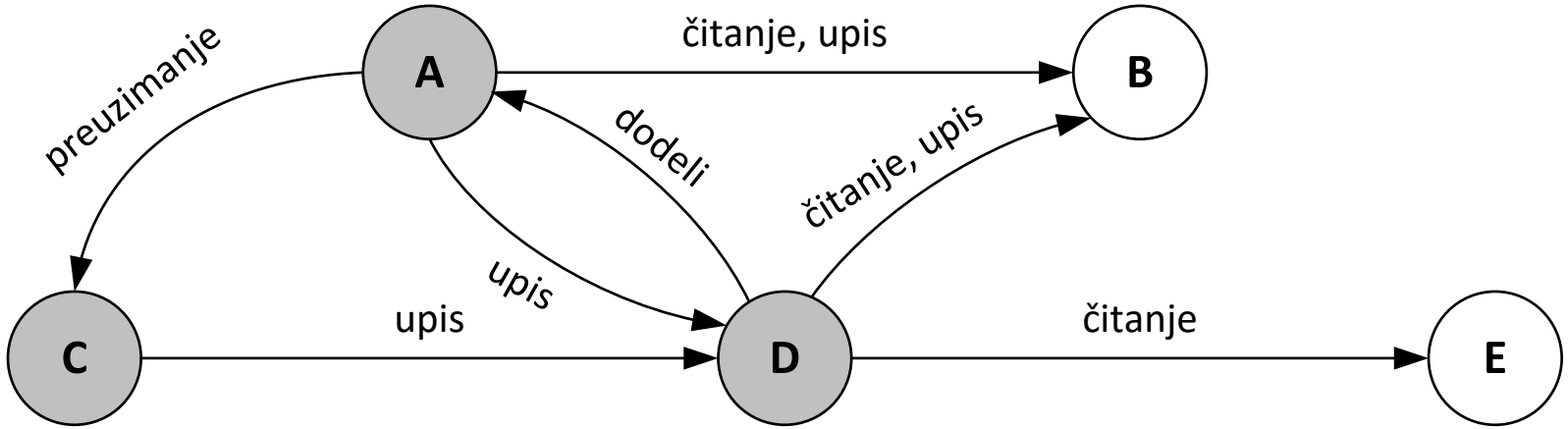
- Kolone matrice pristupa obično se zovu **liste kontrole pristupa** (engl. *access control lists*).
- Vrste matrice obično se se nazivaju **liste sposobnosti** (engl. *capability lists*).
- Model matrice pristupa podržava **diskrecionu kontrolu pristupa**.
  - Elementi u matrici u diskrecionom vlasništvu osobe ili grupe osoba, koje mogu da dodele prava pristupa objektima koji su u njihovom vlasništvu, tj. da administriraju delove tabele.
- U matrici kontrole pristupa, sposobnost subjekta može biti definisana tripletom (objekat, prava, slučajan broj).
  - Ovim tripletom se definišu prava subjekta u odnosu na objekat, zajedno sa slučajnim brojem koji se koristi kao zaštita od lažiranja izvora tripleta.
  - Ovakvi tripleti donekle su slični Kereberos tiketima.

Model preuzmi-dodeli

- Model preuzmi-dodeli zasnovan je na modelu matrice pristupa, s tim što je kontrola pristupa predstavljena **usmerenim grafovima**.
  - **Čvorovi grafa** mogu biti subjekti (na primer, procesi) i objekti (na primer, resursi).
  - **Grana** usmerena od čvora A ka čvoru B znači da subjekat A ima neko pravo nad subjektom (ili objektom) B.
- Grana se obeležava skupom prava koje A ima nad B.
- Model preuzmi-dodeli nudi četiri različita prava pristupa:
  - **Čitanje** (engl. *read*) omogućava čvoru A da pristupi čvoru B bez mogućnosti izmene sadržaja.
  - **Upis** (engl. *write*) omogućava čvoru A da nešto upiše u čvor B.
  - **Preuzimanje** (engl. *take*) omogućava čvoru A da preuzme prava pristupa koja čvor B ima nad nekim drugim subjektom ili objektom.
  - **Dodela** (engl. *grant*) omogućava čvoru A da svoja prava pristupa nad nekim drugim subjektom ili objektom prenese čvoru B.

Model preuzmi-dodeli

- Čvorovi grafa A, C i D su subjekti, a čvorovi B i E su objekti.
- U tabeli dat je model matrice pristupa koji odgovara trenutnom stanju.



subj. / čvor	A	B	C	D	E
A		čitanje, upis	preuzimanje	upis	
C				upis	
D	dodela	čitanje, upis			čitanje

Model preuzmi-dodeli

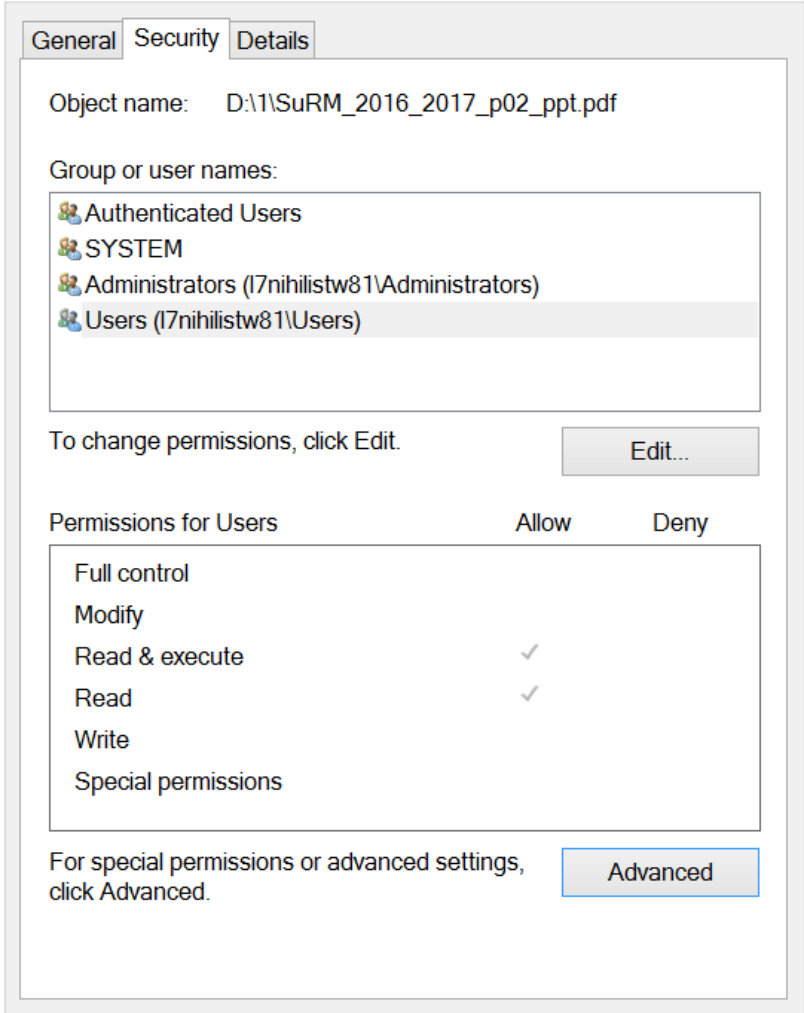
- Pored čvorova i grana (tj. subjekata, objekata i prava), ovaj model definiše i neka pravila koja dozvoljavaju pravljenje i uništavanje čvorova i grana u grafu.
- Grane grafa se mogu dodati i ukloniti primenom prava preuzimanja i dodele.
- Pomoću pravila **napravi** (engl. *create*) može se dodati nov čvor u graf.
  - Ukoliko subjekat A pravi čvor B, u graf se osim čvora B dodaje i grana A-B koja sadrži potpuni skup prava pristupa.
  - To znači da subjekat koji pravi neki čvor ima sva prava nad njim.
- Pomoću pravila **ukloni** (engl. *remove*) mogu se ukloniti određena prava iz neke grane grafa.
  - Ukoliko su iz neke grane uklonjena sva prava u odnosu na neki čvor, grana se takođe uništava.

Pitanje

- Da li Vam je poznat dijalog sa slike?
- Da li Vam je poznat popis sledećih komandi u Linux-u?

```
$ chmod 750 myscript.sh
$ chown nmacek myscript.sh
```

- Da li Vas slika i popis komandi više podsećaju na:
  - model matrice pristupa ili
  - model preuzmi-dodeli?



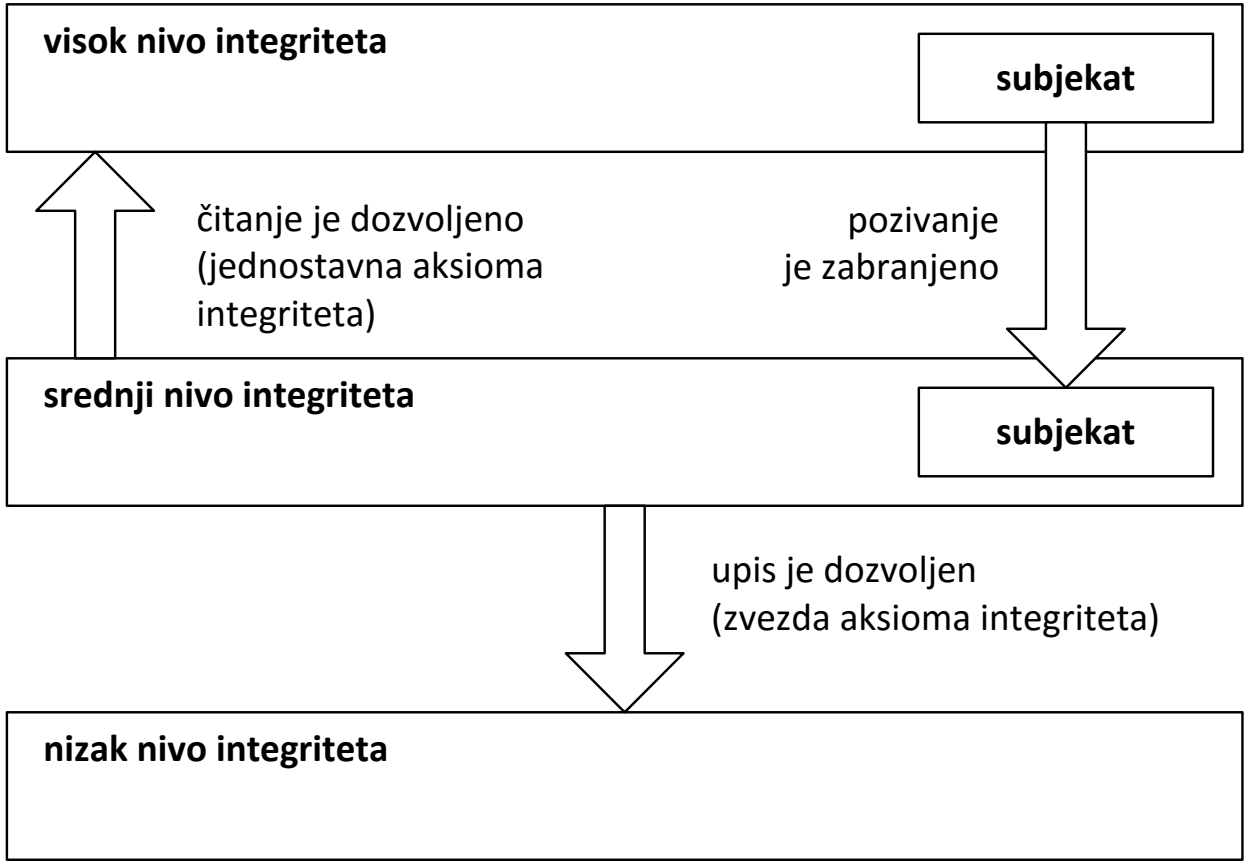
Modeli integriteta

- U mnogim državnim i komercijalnim organizacijama, za određene aplikacije, integritet podataka je važniji nego poverljivost.
- Zbog toga su razvijeni formalni modeli integriteta.
- Prvi model razvijen je kao analogija Bell-LaPadula modelu poverljivosti, a nakon toga je postao sofisticiraniji kako bi zadovoljio dodatne zahteve koji se odnose na integritet.
- Integritet se može okarakterisati sledećim ciljevima:
  - Podaci su zaštićeni od izmene koju pokušava da obavi neovlašćen korisnik.
  - Podaci su zaštićeni od neovlašćene izmene koju pokušava da obavi ovlašćeni korisnik.
  - Podaci su interno i eksterno konsistentni.
  - Na primer:
    - Podaci koji se čuvaju u bazi podataka moraju biti u skladu sa pravilima baze podataka (moraju biti zadovoljena pravila integriteta entiteta i referencijalnog integriteta).
    - Podaci moraju verodostojno oslikavati spoljašnjost, tj. realni svet koji je modeliran tom bazom podataka.

Biba model integriteta

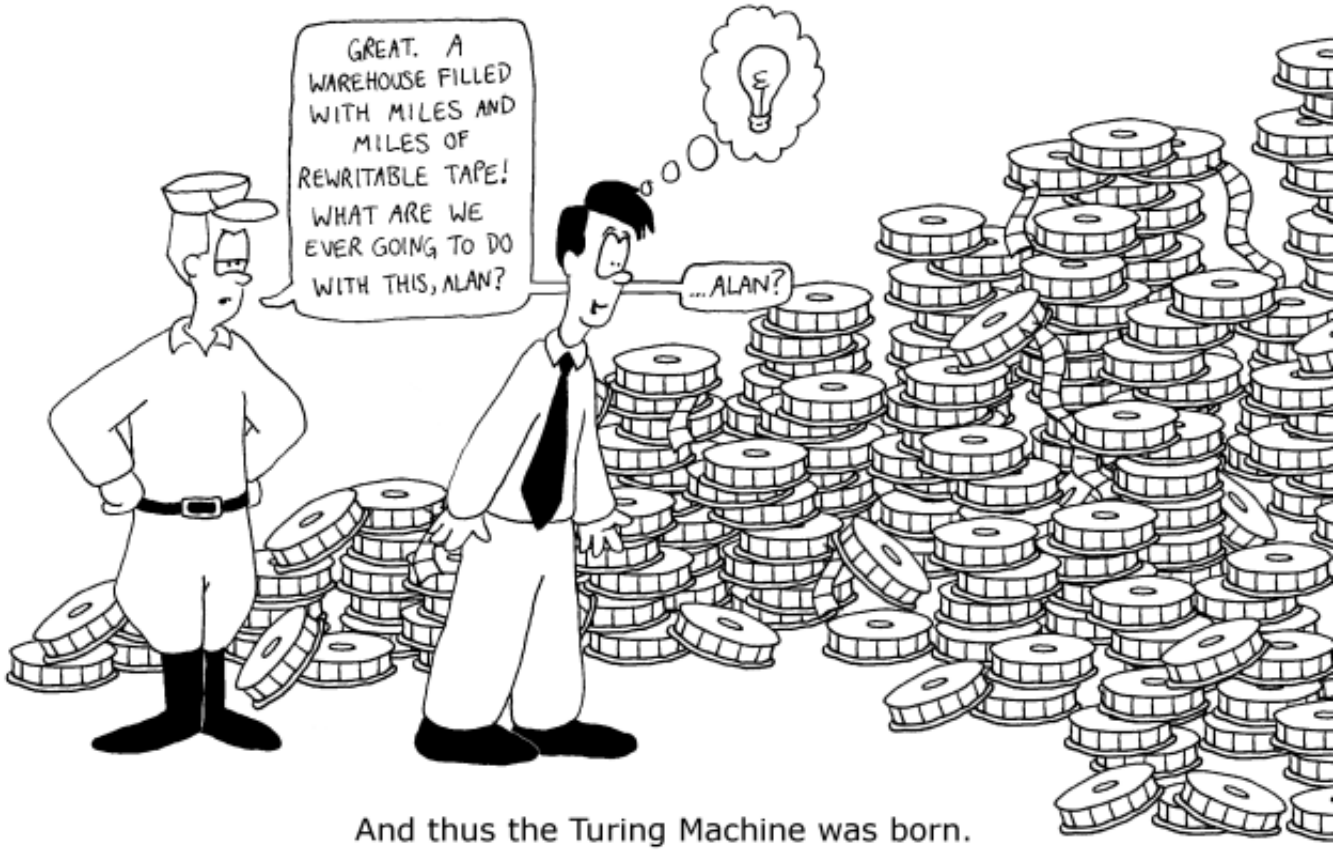
- Slično klasifikaciji različitih nivoa osetljivosti u Bell-LaPadula modelu, Biba model klasifikuje objekte u različite **nivoe integriteta**.
- Model specificira sledeće tri aksiome integriteta:
  - **Jednostavna aksioma integriteta** (engl. *simple integrity axiom*). **Nema čitanja nadole** (engl. *no read down*). Subjektu na jednom nivou integriteta nije dozvoljeno da vidi (čita) objekat nižeg integriteta.
  - **Zvezda (*) aksiom integriteta** (engl. *star integrity axiom*). **Nema pisanja prema gore** (engl. *no write up*). Objektu na jednom nivou integriteta nije dozvoljeno da izmeni, tj. modifikuje objekat višeg nivoa integriteta niti da upisuje u njega.
  - Subjekat na jednom nivou integriteta ne može pozivati (engl. *invoke*) subjekat na višem nivou integriteta.
- Napomena: Biba model je takozvani *lattice-based* model i koristi relaciju manje od ili jednako.
  - Mrežna ili rešetkasta struktura definisana je kao delimično uređen skup sa najmanjom gornjom granicom i najvećom donjom granicom.
  - Mreža predstavlja skup klasa integriteta i uređenu relaciju između njih.

Biba model integriteta



Neformalni opis Tjuringove mašine

Neformalni opis Tjuringove mašine



And thus the Turing Machine was born.

* Slika preuzeta sa stranice: http://adventofcomputers.weebly.com/alan-turing-the-father-of-computing.html

Neformalni opis Tjuringove mašine

- Tjuringova mašina se sastoji od:
  - **Beskonačne trake** koja je podeljena na **ćelije**. Sadržaj svake ćelije može biti 0 ili 1.
  - **Glave** koja se nalazi nad tačno jednom ćelijom. Glava može čitati sadržaj svake ćelije, upisivati 1 ili 0 u ćeliju, pomerati se jednu ćeliju ulevo ili udesno.
  - **Indikatora stanja**.
- Tjuringova mašina se u svakom trenutku nalazi u jednom od konačno mnogo stanja.
- Skup svih stanja se obeležava sa: $S = \{q_0, q_1, ...\}$
- U svakom koraku mašina analizira stanje u kojem se nalazi i sadržaj ćelije nad kojom je glava.
- Mašinom upravlja program koji je sačinjen od konačnog niza naredbi oblika $q_i\ s\ o\ q_j$ gde su: s znak nad kojim se nalazi glava, $o \in \{1, 0, L, R\}$ oznaka operacije:
  - Ako je $o = 1$, u ćeliju nad kojom se nalazi glava upisuje se 1
  - Ako je $o = 0$, briše se sadržaj ćelije nad kojom se nalazi glava (upisuje se 0)
  - Ako je $o = L$, glava se pomera jednu ćeliju ulevo
  - Ako je $o = R$, glava se pomera jednu ćeliju udesno.
- Nakon izvršenja naredbe, mašina prelazi u stanje q_j.

Neformalni opis Tjuringove mašine

- Primeri:
  - q_5 0 1 q_{17}
    - Ako se mašina nalazi u stanju q_5 , a glava nad znakom blanko, u ćeliju se upisuje znak 1 i prelazi u stanje q_{17}.
  - q_1 0 0 q_2
    - Ako se mašina nalazi u stanju q_1 , a glava nad znakom blanko, u ćeliju se upisuje blanko znak i prelazi u stanje q_2. Ovakva naredba služi samo za promenu stanja mašine.
  - q_0 1 L q_0
    - Ako se mašina nalazi u stanju q_0, a glava nad znakom 1, glava se pomera ulevo, a mašina ostaje u istom stanju.
- Da bi mašina radila **deterministički**, program ne sme sadržati više od jedne naredbe za svaku kombinaciju stanja q_i i sadržaja s ćelije, kao na primer: q_1 1 1 q_2 i q_1 1 L q_3.
- U slučaju **nedeterminističkih** mašina ovaj zahtev ne postoji.

Neformalni opis Tjuringove mašine

- Konvencije:
  - Stanje $q_0 \in S$ je **početno stanje**. Mašina se inicijalno nalazi u početnom stanju.
  - Traka sadrži konačno mnogo ćelija u koje je upisan znak 1, dok ostale sadrže znak 0.
  - **Reč** se na traci prikazuje kao neprekidan niz ćelija koje sadrže znak 1, a sa obe strane reči postoji bar po jedan blanko znak.
  - Na početku i na kraju programa, glava se nalazi iznad **prve** (najlevlje) ćelije koja sadrži 1.
  - Mašina prekida sa izvršavanjem kada se nađe u **završnom stanju** q_z.
- Pod **konfiguracijom** Tjuringove mašine podrazumevamo opis koji sadrži: opis sadržaja trake, položaj glave i stanje mašine.
- Standardna konfiguracija u kojoj je:
  - Traka prazna (tj. sve ćelije sadrže blanko znak) ili sadrži najviše konačno mnogo nepraznih reči razdvojenih po jednim blanko znakom.
  - Glava mašine je iznad prve ćelije trake koja sadrži znak 1.
  - Ako počinje sa izvršavanjem, mašina se nalazi u početnom stanju q_0, a ako završava sa radom u završnom stanju q_z.

Neformalni opis Tjuringove mašine

- Na osnovu izloženog, programi se mogu shvatiti kao funkcije koje preslikavaju skup konfiguracija mašine u samog sebe!
- Šta se dešava ukoliko se glava nađe iznad ćelije za čiji sadržaj ne postoji naredba?
- Ova situacija bi odgovarala zaglavljivanju programa pisanih na standardnim programskim jezicima i može se formalizovati kompletiranjem programa naredbama koje u takvim situacijama ne menjaju ni stanje, ni poziciju glave, ni sadržaj ćelije nad kojom se glava nalazi.
  - Na primer, naredba $q_5\ 0\ 0\ q_5$ predstavlja beskonačnu petlju.
- **Tjuring-izračunljivih** funkcija ima **prebrojivo mnogo**.
  - Svaki program je konačan niz naredbi.
  - Svaka naredba je konačan niz simbola iz nekog prebrojivog skupa.
  - Postoji prebrojivo mnogo programa.
- Postoje funkcije koje nisu Tjuring-izračunljive i njih ima neprebrojivo mnogo.

Literatura

1. D. Pleskonjić, N. Maček, B. Đorđević, M. Carić (2007): Sigurnost računarskih sistema i mreža. Mikro knjiga, Beograd.
2. Z. Ognjanović, Nenad Krdžavac (2004): Uvod u teorijsko računarstvo.

Hvala na pažnji

Pitanja su dobrodošla.