

Operacijski sistemi



Operacijski sistem

Vsebina

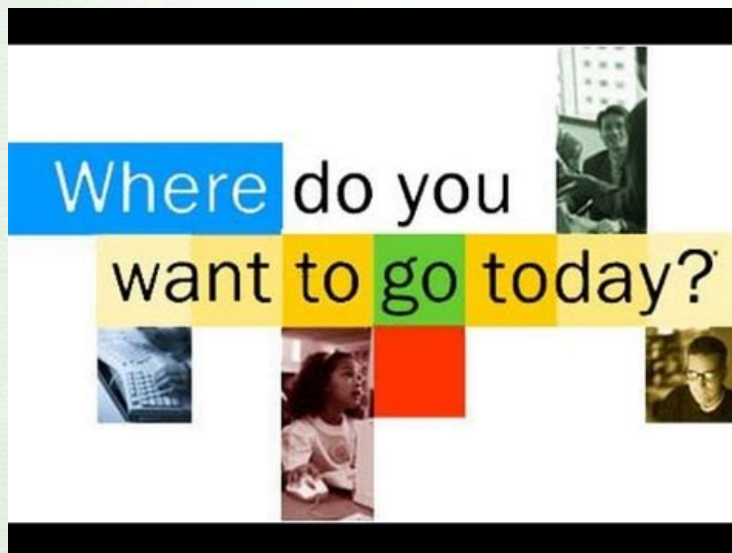
- Operacijski sistem
- Koncepti OS
- Jedro OS
 - Kaj je jedro?
 - Procesorski nivoji zaščite
 - Komunikacija z jedrom
- Arhitektura jedra

Operacijski sistem

- Kaj je operacijski sistem?
 - nabor programske opreme (programski sistem)
 - nadzoruje izvajanje programov
 - povezuje uporabnika s strojno opremo
 - deluje kot vmesnik med programi in strojno opremo

Operacijski sistem

- Kako na OS gleda uporabnik?
 - uporabniški pogled
 - vloga OS, kakor jo vidi uporabnik računalnika
 - OS kot vmesnik uporabnik-računalnik
 - cilj: kar se da olajšati uporabo računalnika

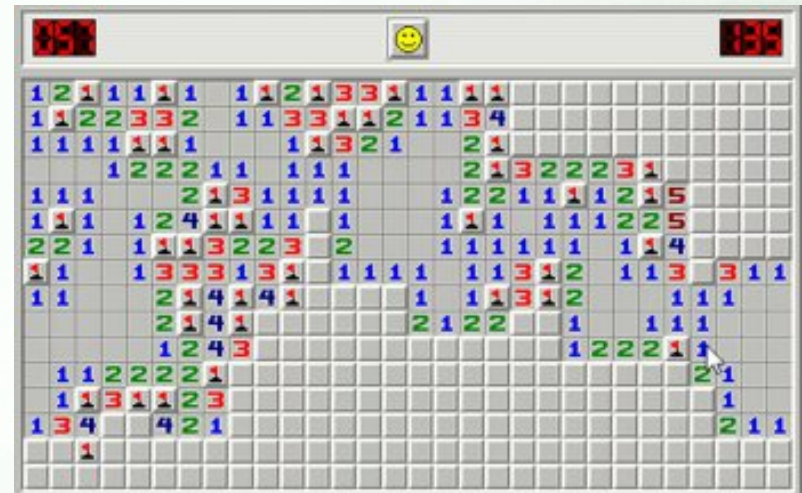


Operacijski sistem

- Kako na OS gleda računalničar?
 - sistemski pogled
 - vloga OS, kakor jo vidi računalničar, programer ipd.
 - OS kot upravljalec računalniških virov
 - npr. procesor, pomnilnik, datoteke
 - OS kot nadzornik delovanja sistema
 - nadzornik delovanja programov in strojne opreme
 - OS kot ponudnik (sistemskih) storitev
 - npr. dostop do datotek, uporaba naprav

Operacijski sistem

- Kaj sodi pod operacijski sistem?
 - jedro, gonilniki naprav, lupina, sistemska orodja
 - v praksi OS pogosto vsebuje tudi neko SPO
 - odvisno od proizvajalca
 - GNU/Linux
 - Linux ... jedro OS
 - GNU ... programska oprema



Safari



Mail



Contacts



Calendar



Reminders



Notes

Operacijski sistem

- Katere storitve nudi operacijski sistem?
 - upravljanje z uporabniki
 - upravljanje s procesi
 - upravljanje s pomnilnikom
 - upravljanje z datotečnimi sistemi in datotekami
 - upravljanje z V/I napravami
 - medprocesna komunikacija in sinhronizacija
 - računalniška komunikacija in mreženje
 - zaščita procesov in varnost podatkov
 - statistika in računovodstvo (o uporabi virov)
 - ...

Operacijski sistem

- Kaj so cilji operacijskih sistemov?
 - prikladnost uporabe (ease of use)
 - računalnik je preprosto uporabljati
 - varnost uporabe (security)
 - uporaba računalnika je varna
 - zanesljivost delovanja (reliability)
 - računalnik deluje brez izpadov in po pričakovanjih
 - učinkovitost delovanja (performance)
 - računalnik je dobro izkoriščen in hitro deluje
 - sistemski prog. jeziki: C, C++, D, Rust
 - prilagodljivost sistema (flexibility)
 - ustvarja okolje za nove funkcionalnosti

Veseli bodimo, da nam ni treba ves čas uporabljati zbirnika.



Operacijski sistem

- Zakaj študirati operacijske sisteme?
 - uporabnik
 - vsakodnevna uporaba računalnika poteka preko OS
 - aplikacijski programer
 - boljše zmogljivosti aplikacij, manjša poraba virov
 - sistemski programer
 - poznavanje OS je nujno
 - računalničar
 - poznavanje OS je temeljno računalniško znanje
 - teoretični računalničar (computer scientist)
 - za celovito razumevanje računskih strojev

Koncepti operacijskih sistemov

- Abstrakcija (abstraction)
- Virtualizacija (virtualization)
- Varnost (security)
- Sočasnost (concurrency)
- Persistenca (persistence)

Splošni koncepti,
ki se pojavljajo pri
študiji OS.



Koncepti operacijskih sistemov

- Abstrakcija

- posplošitev in skrivanje podrobnosti
- omogoča poenotenje in združevanje podobnih entitet v eno krovno entiteto
- primer: datoteka
 - abstrakcija datoteke kot hranjene zbirke podatkov
 - fizično je datoteka lahko hranjena na veliko različnih načinov

Koncepti operacijskih sistemov

- Virtualizacija

- mehanizem, ki nekaj ustvari navidezno
 - npr. navidezna naprava, procesor, pomnilnik
- preslikava navideznega v realno
 - vmesnik in viri navidezne naprave se preslikajo na vmesnik in vire realne naprave
- podrobnosti lahko pusti odprte
- primer: navidezni procesor
 - emulacija procesorja Z80
- primer: navidezni pomnilnik

Koncepti operacijskih sistemov

- Abstrakcija in virtualizacija
 - komplementarna koncepta
 - primer: navidezni datotečni sistem (VFS)
 - nudi enovit dostop do datotek
 - sistemski klici: open, read, write, close
 - združuje različne naprave
 - preslikuje datotečni sistem na posamezne naprave
 - vključuje različne datotečne sisteme
 - fizični datotečni sistemi različnih naprav (ext2, NTFS, ...)
 - psevdo datotečni sistemi (procfs)
 - mrežni datotečni sistemi (AFS)

Koncepti operacijskih sistemov

- Varnost

- zaupanje v "dobro" delovanje sistema
- dosežemo jo preko mehanizmov zaščite sistema
- primeri:
 - izolacija procesov
 - zaščita pomnilnika
 - zaščita V/I naprav
 - zaščita datotek

Koncepti operacijskih sistemov

- Sočasnost
 - obstoj več procesov hkrati
 - občutek hkratnega izvajanje več procesov
 - težave in izzivi uporabe sočasnosti
 - sodelovanje procesov in njihova sinhronizacija
- primer:
 - procesi, kot neodvisne izvajalne enote
 - niti, kot tesno sodelujoče izvajalne enote

Koncepti operacijskih sistemov

- Persistenca

- dolgoročni obstoj podatkov in informacij
 - mehanizmi kot beleženje dnevnika (journaling)
- učinkovitost hrambe
 - kopiraj-ob-pisanju (copy-on-write)
- omogoča tudi deljenje informacij
 - medprocesna komunikacija
- primer: datoteka

Jedro

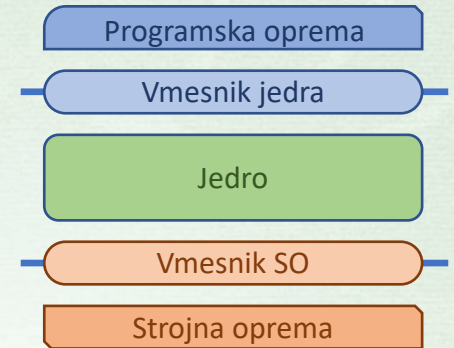
- Kaj je jedro OS?
 - programska koda, ki vsebuje bistveni del OS
 - vsebuje najpomembnejše storitve OS
 - npr. upravljanje s procesi in pomnilnikom
 - se izvaja v privilegiranem načinu delovanja procesorja
 - lahko obvladuje celoten sistem
 - privilegirani strojni ukazi, rokovanje prekinitev
 - nastavljanje sistemskih registrov
 - neposreden dostop do V/I naprav
 - brez tega načina ni mogoče narediti modernega OS

Jedro



Jedro

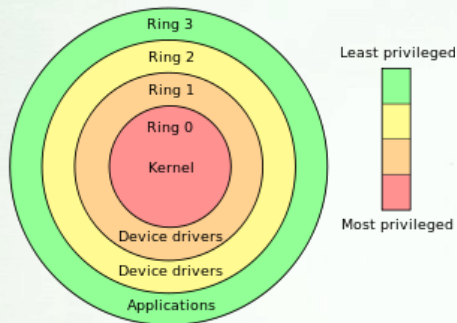
- **Procesorski nivoji zaščite**
 - **uporabniški prostor** (zaščiteni način)
 - omejena uporaba procesorja
 - nadzorovan dostop do pomnilnika, neposreden dostop do naprav ni mogoč, nekaterih ukazov ni mogoče izvesti
 - napačna uporaba povzroči izjemo
 - **jedrni prostor** (privilegirani način)
 - neomejen dostop do pomnilnika in naprav
 - nekateri ukazi se lahko izvajajo samo v tem načinu
 - npr. prekinitve, pasti, vpliv na (pred-)pomnilnik
 - x86: sti, cli, int x, *flags*
 - arm: swi, mrs, msr
 - **hipervizorski način**
 - za samo izvedbo OS ni potreben
 - je pa potreben za virtualizacijo



Jedro



- *Procesorski nivoji zaščite
 - arhitektura x86 – obroči zaščite



x86 ima 4+3 obroče

3 ... zaščiteni način

2 ... se ne uporablja / programi s privilegiji dostopa do naprav

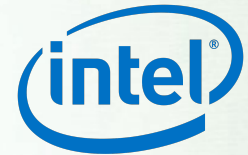
1 ... se ne uporablja

0 ... privilegirani način

-1 ... hipervizorski način

-2 ... SMM, system management mode

-3 ... Intel Management Engine, AMD Platform Security Processor



- arhitektura ARMv7 (PL – privilege level)
 - PL0 – uporabniški nivo
 - PL1 – jedrni nivo
 - PL2 – hipervizorski nivo



Jedro

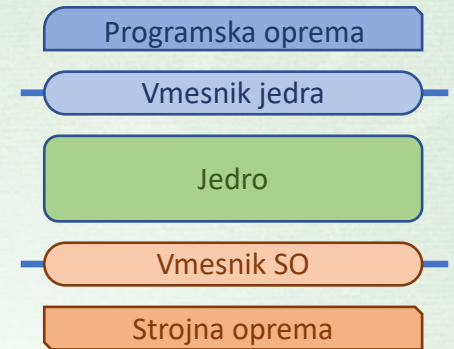
- **Komunikacija med jedrom in strojno opremo**

- **naprava**

- dejanska naprava, ki omogoča neko opravilo
 - npr. tipkovnica, miška, trdi disk, mehki disk, SSD disk, USB ključ, grafična kartica, monitor, ...

- **kontrolniki naprav**

- elektronska vezja, ki razumejo ukaze, podane na vmesniku SO, in jih ustrezno posredujejo napravi
 - npr. kontrolnik USB, kontrolnik SATA, ...



Jedro

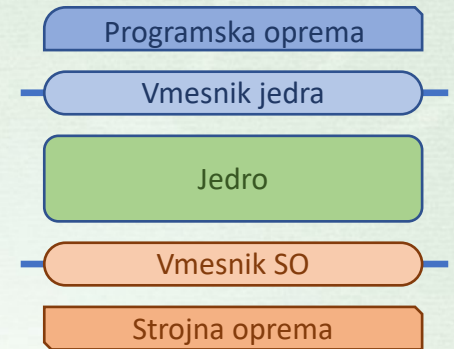
- Komunikacija med jedrom in strojno opremo

- **vmesnik strojne opreme**

- mehanizem programskega podajanja ukazov napravam
 - funkcionalnost strojne opreme
 - npr. pomnilniško preslikan V/I, ločen V/I naslovni prostor, DMA – neposreden dostop do pomnilnika, strojne prekinitve

- **gonilniki naprav** (device drivers)

- programska koda, ki zna upravljati z neko napravo preko vmesnikov strojne opreme
 - navadno so gonilniki del jedra



Jedro

- Komunikacija med programsko opremo in jedrom

Programska oprema

Vmesnik jedra

Jedro

Vmesnik SO

Strojna oprema

Kako pa programi zahtevajo storitve jedra?

Preko vmesnika jedra.

Kaj pa je vmesnik jedra?

Pravzaprav gre za systemske klice.

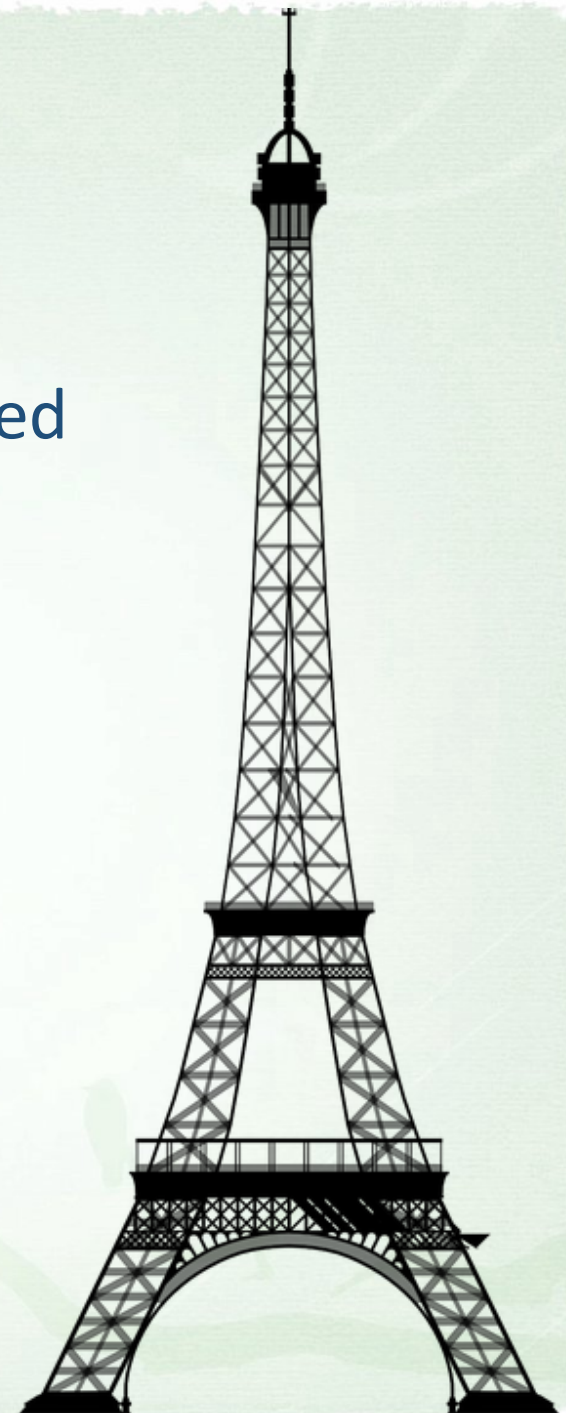
Uhh, ok? Za klice sistema?

*Da, za nekaj podobnega kot so klici funkcij.
Podrobnosti si bomo pogledali kmalu.*



Arhitektura jedra

- Kaj je arhitektura jedra?
 - **struktura** in **način povezovanja** med posameznimi deli jedra
- Vrste jedrnih arhitektur
 - monolitno jedro
 - monolitno modularno
 - mikro jedro
 - hibridno
 - unijedro
 - ostalo



Arhitektura jedra

- Monolitno jedro
 - (velik) kos strojne kode
 - teče v jedrnem nivoju v skupnem naslovnem prostoru
 - vsebuje celoten OS (podsistemi, gonilniki)
 - hitrost
 - posamezni deli OS lahko med seboj zelo hitro komunicirajo, preko navadnih klicev funkcij
 - slabosti
 - napaka v enem delu OS lahko povzroči sesutje celotnega OS
 - težja obvladljivost programske kode
 - sprememba izvirne kode → ponovno prevajanje jedra
 - primeri
 - System V, DOS, Windows 9x, FreeDOS, OpenVMS

Arhitektura jedra

- Monolitno modularno jedro
 - omogoča modularno zasnovo jedra (navkljub monolitnosti)
 - modul navadno vsebuje gonilnik naprave
 - dinamičnost
 - module je moč vložiti (load) in tudi izločiti (unload) iz jedra tekom samega izvajanja (ni potreben reset sistema)
 - primeri
 - BSD, Linux, OS-9, AIX, ...

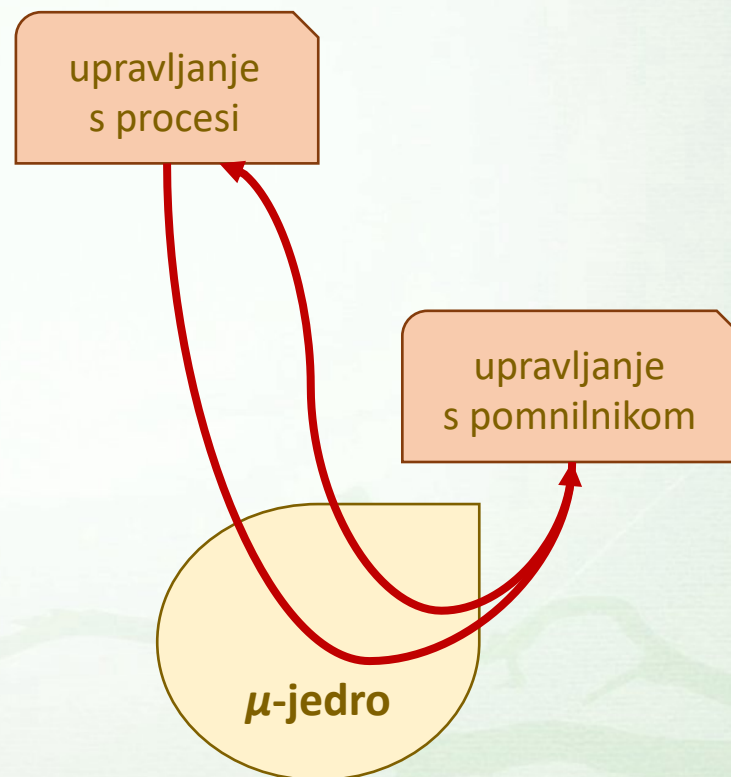
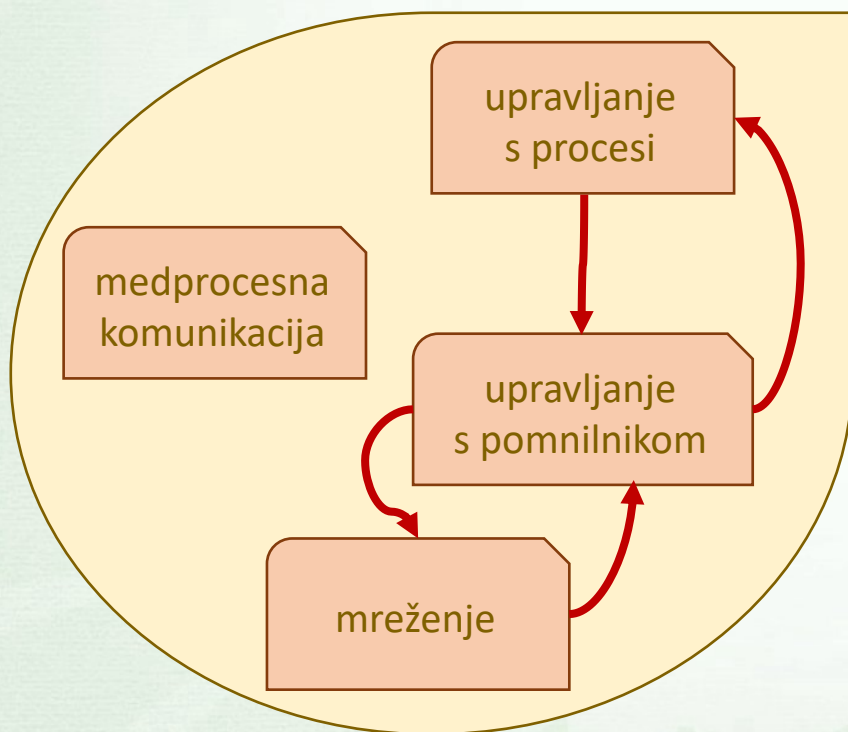
Arhitektura jedra

- **Mikro jedro (μ -jedro)**

- vsebuje samo osnovne funkcionalnosti
 - procesi, pomnilnik, medprocesna komunikacija
- ostale funkcionalnosti so izvedene preko procesov (v uporabniškem načinu)
 - uporaba medprocesne komunikacije
 - princip komunikacije: odjemalec-strežnik
 - medsebojni klici so časovno zahtevnejši
- prilagodljivost, porazdeljenost, varnost, enostavnejša implementacija

Arhitektura jedra

- Monolitno vs mikro jedro

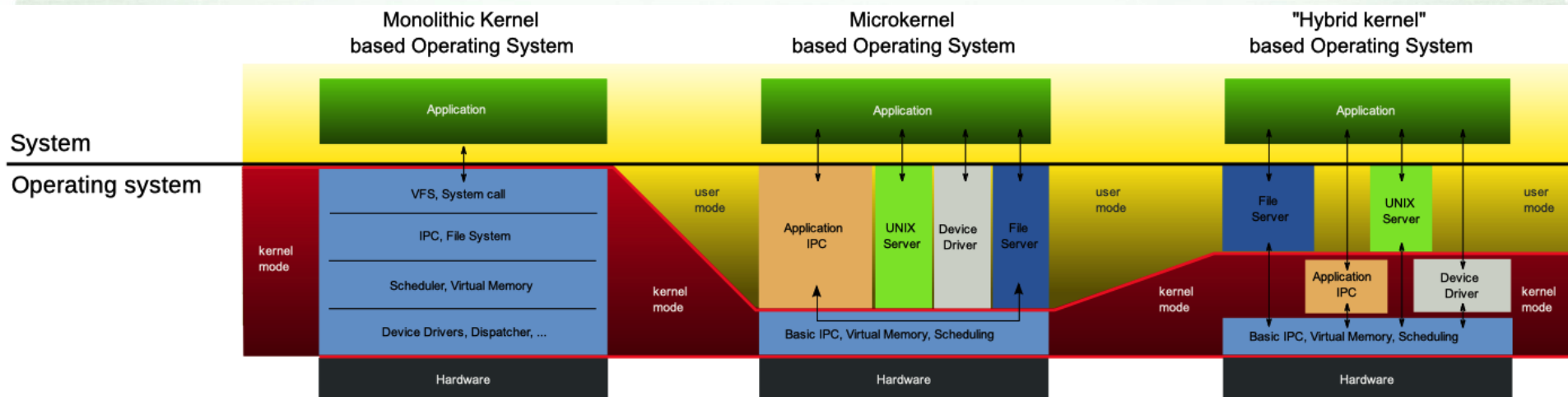


Arhitektura jedra

- Hibridno jedro
 - zasnova je mikro jedrna
 - ločeni servisi
 - izvedba pa je monolitna
 - skupen naslovni prostor
 - primer
 - Windows NT

Arhitektura jedra

- Primerjava: monolitno, mikro in hibridno jedro



Arhitektura jedra

- *Arhitektura jedra – ostale vrste
 - **nano jedro, piko jedro**
 - različni pomeni: enako kot mikro jedro, manjše kot mikro jedro, hipervizor, HAL (hardware abstraction layer)
 - **exokernel**
 - še manjši od mikro jedra, omogoča le zaščito in souporabo virov
 - **unikernel**
 - specifično namensko jedro za izbrano aplikacijo
 - sestavljeno iz izbranih komponent
 - podobni monolitnim: velika koda, enotni naslovni prostor
 - navadno se izvaja v virtualnem okolju