

1. Opišite naloge jedra pri sistemu UNIX. Kaj so naloge jedra in kaj naloge lupine ob izvršitvi ukaza rm moja_datoteka?

Naloge jedra so:

- nadzor nad procesi, rojevanje in pobijanje procesov, upravljanje z resursi sistema
- oskrba procesov s podatki, zagotavljanje navideznega stroja (edinega procesa) za proces
- sprejema strojne prekinitve periferije, delo z datotekami, delo z virtualnim spominom, mrežni protokoli, večopravilnost (niti)

Ko uporabnik napiše ukaz (rm datoteka) v ukazni vrstici, lupina kreira (fork) nov proces (klon lupine), poišče kodo za ukaz rm, skopira kodo preko kode kloniranega procesa in kernelu sporoči, da je nov proces pripravljen za izvedbo. Osnovnemu procesu rečemo "oče", novemu pa "otrok".

2. Razložite vlogo sistemskih klicev na sistemu UNIX. Naštejte bistvene razlike pri delu z datotečnim sistemom z uporabo sistemskih klicev in pri delu z datotečnim sistemom z uporabo funkcij knjižnice.

V UNIXu vsi uporabniški programi in aplikacije uporabljajo vmesnik sistemskih klicev za dostopanje do sistemskih virov kot so diski, tiskalniki, spomin,... Vmesnik sistemskih klicev v UNIXu ponuja množico sistemskih klicev (C funkcij). Naloga vmesnika sistemskih klicev je zagotoviti celovitost in nedotakljivost sistema (integriteto). Ves dostop do strojne opreme na nizkem nivoju poteka pod kontrolo operacijskega sistema, s tem pa je uporabnikovim programom onemogočeno onesposobljenje sistema.

Operacijski sistem ob sprejemu sistema klica preveri njegovo avtentičnost ali dovoljenje, nato pa ga izvede v imenu uporabniškega programa ter vrne rezultat. Če je zahteva napačna ali pa nedovoljena, sistem zahteve ne izvede in programu vrne kodo napake.

Sistemski klici so dosegljivi kot množica funkcij v Cju, saj je tudi UNIX napisan v tem jeziku

Sistemski klici so edini način za dostop do funkcij jedra kot so datotečni sistem, večopravilni mehanizem, komunikacija med procesi.

3. Razložite pozicijske argumente in vidljivost spremenljiv v lupini bash. Kako je s pozicijskimi parametri ukazne datoteke in kako s pozicijskimi parametri funkcij? Kako je s spremenljivkami ukazne datoteke in kako s spremenljivkami funkcij?

Ko bash skript kliče funkcijo, se argumenti pri klicu funkcije spremenijo v pozicijske parametre. Uporabljamo jih na isti način, kot pozicijske argumente skripte. Argument št.0 (#0 - ime skripte) ostane nespremenjen.

Če se v funkciji izvrši komanda return, se funkcija konča, skripta se nadaljuje takoj za klicem funkcije, pozicijski argumenti pa se vrnejo na prvotno vrednost. Če ukaz return vrne numerično vrednost, je dosegljiv kot izhodni status funkcije, drugače pa je izhodni status funkcije enak statusu zadnjega ukaza pred return.

Spremenljivke v skriptu so globalne. Spremenljivke, ki so lokalne v funkciji, se deklarirajo s predpono local. Te spremenljivke so vidne samo v okviru funkcije, ki jih vsebuje.

4. Primerjajte operacijska sistema Win NT in UNIX. Kaj imata skupnega? V čem se bistveno razlikujeta?

Windows 95 oziroma Windows 98 je v razliko njegovih predhodnikov (ki so sistemu MS DOS dodali le grafični vmesnik), samostojen operacijski sistem. Namenjen je 32-bitnim procesorjem.

Windows NT je enouporabniški, večopravilni operacijski sistem, predvsem namenjen omrežjem, delujočim po konceptu klijent - strežnik. Ima enak grafični vmesnik kot Windows 95. Je plačljiv.

UNIX je večuporabniški, večopravilni operacijski sistem, ki so ga razvili v Bellovih laboratorijih. Njegov prvotni namen je bil predvsem v kompleksnih znanstvenih aplikacijah. V primerjavi z Microsoftovimi produkti ni tako uporabniško prijazen. Je open source.

Podobnost/razlika:

- Podoben datotečni sistem (razlike: en sam root, linki)
- Podobno koncipiran vhodni jezik (cp..copy, ls..dir,cd..cd,..)
- Bolj bogat vhodni jezik (cevi, krmilni ukazi, ukazne datoteke)
- Več tipov lupin sh, csh, ksh, bash, tcsh,..
- Več programskih procesov, proces v ospredju, procesi v ozadju,
- Več uporabnikov, multiscreen,
- Medprocesna komunikacija (cevi,..)
- Večprocesorski sistemi, porazdeljeno računanje
- Večje podatkovne strukture (virtualni pomnilnik, swapping)
- Večje datoteke

7. Razložite pomen in delovanje nalagalnikov, ki jih srečamo pri zagonu operacijskega sistema (Loader, Bootstrap Loader). Kaj je funkcija LILO oziroma Grub?

“Bootstrap loader” locira jedro (operacijskega sistema), ga naloži v pomnilnik in sproži njegovo izvajanje. V nekaterih primerih pa preprosto “bootstrap loader” poišče na disku bolj kompleksen zagonski program, ga naloži v pomnilnik, ta pa nato naloži jedro (kernel).

Bootstrap loader je program, ki naloži “prvi program”. Pogosto “večstopenjski”: primarni, sekundarni. Terja podporo firmware (“hardware bootstrap”). Naš prvi program pa je lahko tudi Loader (nalagalnik) oz. Sekundarni zagonski nalagalnik. LILO oz. Grub sta primer sekundarnega zagonskega nalagalnika. Funkcija teh nalagalnikov je, da nam ponudi Izbiro jeter Linux, Nastavitve časovnih parametrov zagona, Zagon jeder, ki niso Linux, Nastavljanje konfiguracij.

8. Middleware(kaj je to, zakaj se rabi), RPC-kaj je, kje se uporablja...

Programska oprema, ki omogoča aplikaciji ali komponentam aplikacije povezovanje in izmenjavo podatkov z drugimi aplikacijami/komponentami. Programer in uporabnik ne potrebujeta poznavanja internega procesiranja za doseganje teh ciljev. Programer uporablja visokonivojske API (Application program interface). Dodatne značilnosti:

- Skrivanje heterogenosti in prenosljivost
- Lokacijska transparentnost - aplikacija najde storitev na omrežju, ne glede na to kje se le-ta nahaja
- Neodvisnost od omrežne arhitekture
- Zanesljivost
- Skalabilnost

Primer middleware je RPC. RPC – Remote Procedure Calls (RPC). Aplikacije kličejo procedure, ki tečejo na oddaljenih računalnikih, sinhrono ali asinhrono. Značilnosti:

- Podatke prenašamo kot komunikacijske parametre.
- Vhodni parametri morajo ubogati semantiko klica po vrednosti.
- V porazdeljenem smislu nimamo globalnega konteksta.
- Naslavljanje pomnilnika s kazalci nima smisla.

6. a) Imamo dva programa, ki tečeta na različnih računalnikih, povezanih med seboj v mreži. Kako lahko program A pošlje programu B polje n celoštevilčnih podatkov? Povejte le, kaj veste o primernem principu take komunikacije. b) Kaj pa pride v poštev, če programa tečeta na različnih računalnikih?

a) Sockets (vticnice) – V skripti sicer uporabljeno kot izmenjava podatkov med procesi ampak mislim, da veljajo tudi tukaj. Vsak socket je mapiran (mapped) na aplikacijo ali proces. Socket je povezovalnik / vmesnik med procesom in TCP/IP protokolnim skladom operacijskega sistema.

b) cevi (pipes), ce sta na istem računalniku, sockets ce v mreži

6. Razložite mehanizem nastanka in mehanizem končanja procesa, ki teče v ospredju na operacijskem sistemu UNIX. Kateri sistemski klici se uporabijo?

Mehanizem nastanka in končanja procesa

Pri inializaciji sistema jedro generira nekaj začetnih procesov (pid0 - scheduler; pid1 - init). Ostali procesi so generirani z dvema osnovnima sistemskima klicema za oblikovanje okolja procesov, fork() in exec().

fork() – naredi dvojnika procesa; stari proces – prehodnik; novi proces – naslednik

exec() – zamenja sliko procesa z drugim procesom exit – konča z delom (vrne predhodnemu procesu status procesa)

wait - sistemski klic omogoča čakanje predhodnika, da njegov naslednik zaključi z delom

kill – ubije proces (sigkill = exit; Sigstop = stop, sigcont)

Fora je v temu, da mu jedro nameni več resorsu. Namreč procesi, kateri so v ozadju imajo prednost pred tistimi, katerimi so v ozadju. Se pravi proces, ki je v ospredju mu je namenjen več pomnilnika, hitrosti, kot tistemu, kateri je v ozadju. Nekako tako! Ima prednost pred ostalimi, saj je trenutno aktualen.

Sam kle bi jest se omenu sirote, ce se predhodni proces konca pred naslednikom naslednik postane sirota in pa pa ce naslednik konca pred predhodnikom predhodnik postane zombie.

6. Naštejte vsaj tri zlonamerne kode in opišite razlike med njimi. (Virusi, črvi,...)

Trojanski konji:

Trojanski konj oponaša funkcionalnost legitimnega istoimenskega programa. Lahko je vključen v nek legitimni program, lahko je samostojen program. Pomembno je, da imam dodatno skrito poslovanje. Ne more delovati sam

od sebe, uporabnik ga mora pognati. Najbolj pogoste funkcije: brisanje ali prepisovanje podatkov, upload/download datotek, dovoljevanje oddaljenega dostopa

Črvi:

Črvi so programi, ki se po omrežju širijo od računalnika do računalnika, tečejo samostojno, nasprotju z virusi se ne navezujejo na obstoječe datoteke, imajo lahko (različne) dele svojih kopij, ki tečejo na več različnih računalnikih. Najbolj pogoste funkcije: Tipično škodijo (preobremenjujejo) omrežjem, nosijo druge naloge, brisanje datotek, odpiranje zadnja vrata.

Virusi:

“Programi”, ki se skrivajo v drugih programih/datotekah v računalniku, znajo reproducirati in izvajati brez vednosti uporabnikov, potrebujejo gostiteljski program in ko teče gostiteljski program teče tudi virus. Najbolj pogoste funkcije: zlonamerno škodo (na primer zbrise tabelo particij), Vpišejo se na področje zagonskega sektorja medija in trdega diska, se širijo preko elektronske pošte kot priponka.

10. Kakšne možnosti imamo pri sistemu LINUX, da restavriramo pomotamo izbrisane datoteke.

Ena možnost kako lahko restavriramo pomotamo izbrisano datoteko je preko programa, ki ima še po možnost odprto link na to datoteko. Tukaj pride v poštev /proc direktorij, kjer ima vsak proces svoj direktorij, ki vsebuje tudi fd ("file descriptor"), ki kaže na odprto datoteko. Iz teh podatkov dobimo file descriptor preko katerega potem dobimo datoteko cp /proc/4158/fd/4 myfile.saved. (google)

Druga opcija je, da gremo v single mode način (wall ukaz)in nato uporabimo grep ukaz (primer: grep -i -a -B10 -A100 'nixCraft' /dev/sda1 > file.txt). V single user moramo iti, da kdo drug od uporabnikov ne zasede tega prostora, kjer je trenutno dejansko shranjena naša datoteka, vendar je bil link na to izbrisan. (google)

Obstajata še dva programa za pomoč pri teh zadevah in sicer:

recover
e2undel

Drugi možnost ni in je velika verjetnost, da smo datoteko izgubili.

12. Razložite, kaj pojmuje pod konfiguriranjem operacijskega sistema. Razložite tudi pomen registra (Regitry) pri operacijskem sistemu Windows. Kako ga spreminjate?

hm tole bi blo lahko tm pri utrjevanju sistema, pac kofiguracija kot spreminjanje upo imen, passwordov, odstranjevanje nepotrebnih komponent, dodajanje zaščitnih plasti,...

Register - baza kjer so shranjene nastavitve in opcije OS win. Vsebuje informacije in nastavitve za hardware, OS software, vecino ostalega software, preference,...

dostop z regedit

13. Kaj je skalabilnost sistema in kakšne probleme odraža. Kako je s skalabilnostjo v porazdeljenih sistemih?

Skalabilen sistem je sistem, ki ga lahko enostavno spremenimo tako, da bo dovoljeval povečevanje števila uporabnikov, virov in računalnikov.

14. Razložite pomen particij na trdem disku. Razlozite pojem RAID in pomen njegovih posameznih nivojev. Kaj so zvezki s progo (striped volumes)?

RAID - standard za povezovanje diskov in upravljanja z njimi:

Mirroring (RAID1) - pomembna zanesljivost

Striping (RAID0) - pomembna zmogljivost

Zvezki s progo so RAID 0 (also known as a stripe set or striped volume) splits data evenly across two or more disks (striped) with no parity information for redundancy.

Pomen particij: Večinoma OS dovoli razbijanje diska na več manjših logičnih particij. S tem podamo uporabniku možnost, da si organizira prostor po njegovi lastni voljo, dodatno pa s tem povečamo organizacijo samega sistema. Dodatni razlog za particije je tudi to, da ima manjša particija lahko manjši cluster size, ki določa kakšen je najmanjši podatek, ki ga lahko particija shrani.

7. Narišite diagram prehajanja stanj procesa na večopravilnem sistemu, vsako stanje pa na kratko opišite.

nov -> pripravljen <-prekinitev -> tekoc -> koncan
.....|----->-----uvrstitev----->---|..|
.....|.....|
.....|-----<-----cakajoc-----<-----|

15. Aktivni znaki, vse 3 vrste oklepajev

<, >, & -preusmerjevanje
| - pipe

\$ -pridobitev vsebine spremenljivke
", ', \ -citiranje drugih znakov
() - grupiranje ukazov
[] - v pogojih if stavkov
{ } - pri substitucijah v nizih

14. Mehanizem nastanka in končanja procesa

Pri inializaciji sistema jedro generira nekaj začetnih procesov (pid0 - scheduler;pid1 - init). Ostali procesi so generirani z dvema osnovnima sistemskima klicama za oblikovanje okolja procesov, fork() in exec().fork() – naredi dvojnika procesa; stari proces – prehodnik; novi proces – naslendikexec() – zamenja sliko procesa z drugim procesomexit – konča z delom (vrne predhodnem procesu status procesa)wait - sistemski klic omogoča čakanje predhodnika, da njegov naslednik zaključi z delomkill – ubije proces (sigkill = exit; Sigstop = stop, sigcont

15. Vrste klicov – procesov in potek samih klicov:

Uporabnik - Jedro:
preko sistemskih klincev
Uporabnik - Periferija
preko sistemskih klincev do jedra in nato do periferije
Procesi - Jedro
Procesi jedru dajejo zahteveke, jedro odgovarja z podatki.
Jedro - Periferija
Preko gonilnikov, Interrupti.

13. Kaj so spletne storitve, kateri protokoli se uporabljajo, kako so predstavljeni podatki v spletnih storitvah

Spletne storitve (angl.: Web Services) so programske storitve, ki so ponujene na spletu. Sestavljene so iz funkcij (metod), ki jih lahko kličemo oddaljeno preko spleta. Namenjene so medračunalniški komunikaciji. Njihova funkcionalnost je tako lahko uporabljena s strani drugih programov neodvisno od programskega jezika, v katerem so implementirani ali računalnika, na katerem se izvajajo. Tako omogočajo povezljivost med informacijskimi sistemi (aplikacijami) preko spleta.

Web-based (multi-tier architecture): Odjemalec je brskalnik, ki dostopa (prejema) do informacij, ki so na voljo na dveh ali več strežnikih (spletni strežniki, aplikativni strežniki, podatkovni – baze strežnii. Spletni strežnik servira podatke, aplikativni strežnik generira podatke in jih poda spletnemu strežniku. Podatki so največkrat prebrani iz podatkovnega strežnika.

Podatki so predstavljeni na različne načine, največkrat so za prikaz uporabljene tehnologije kot so XML, HTML, CSS. Do podatkov pa lahko dostopamo tudi preko WEB API oz. preko različnih protokolov kot so HTTP, RSS, REST, SOAP in ostale.

Računalniki na internetu komunicirajo med seboj ali s protokolom TCP (Transmission Control Protocol) ali s protokolom UDP (User Datagram Protocol).

9. Server 2008, kaj je aktivni imenik, kaj vsebuje.... kaj je razlika med domenskim strežnikom in globalnim imenikom(al neki takega). Kaj je šablona uporabniškega konta, zakaj se rabi?

Aktivni imenik je servis uporabljen za hranjenje informacij o strukturi ter virih omrežja na celotni domeni, torej je centralna točka omrežja. Aktivni imenik drži različne objekte, ki padejo v tri različne kategorije: resources

(e.g., printers), services (e.g., email), and users (user accounts and groups). Naloga aktivnega imenika je, da drži informacije o objektih, organizira objekte, nadzira dostop uporabnikov, preverja pravice.

Vsak objekt je predstavljena kot svoja entita, ki ima lastne attribute. Določeni objekti lahko držijo tudi druge objekte (container).

Prednosti aktivnega imenika:

Servis, ki hrani podatke o vseh omrežnih sredstvih

Centralizirano upravljanje omogoča hitro iskanje in dostop do sredstev

Medtem ko vsak domenski krmilnik vsebuje polno repliko za svojo domeno, drži strežnik z globalnim katalogom (global catalogue server) omejeno množico atributov za vse objekte v celotnem gozdu. Na primer:

- Attribute, ki so najbolj pogosto iskani
- Ali potrebni za logiranje

Tako zagotavlja hiter dostop do podatkov za avtentikacijo in druga povpraševanja in iskanja. Klijentu ni potrebno skakati od strežnika do strežnika preko več domen, da bi dobil iskan podatek iz imenika. To je razlika med domenskim strežnikom in globalnim imenikom.

Šablona za uporabniški konto je uporabniški konto, ki vsebuje lastnosti, ki naj bi jih imeli uporabniki s skupnimi zahtevami. Šablone za uporabniški konto večajo učinkovitost tvorbe uporabniških kontov s standardiziranimi konfiguracijami.

14. Konfiguracije dinamičnih diskov na Win 2003

Dinamične diske razpoznavata operacijska sistema Windows 2000 in Windows Server 2003. Terminologija dinamičnih diskov uporablja zvezke (volumes) namesto particij oziroma množic (sets). Pet tipov zvezkov:

- Preprosti zvezki (Simple volumes)
- Speti zvezki (Spanned volumes)
- Zvezek s progo (Striped volumes)
- Zrcaljeni zvezki (Mirrored volumes)
- Zvezki Raid-5

Primer Spet zvezek (Spanned Volume): Do 32 diskov, ki jih obravnavamo kot en zvezek, Uporabno za kombiniranje več manjših delov prostora na disku ali za kombiniranje majhnih diskov, Zvezke, formatirane za NTFS lahko razširjamo, Če eden od diskov spetega zvezka izpade, je nedostopen celoten zvezek, Če zbrisemo del spetega zvezka, je zbrisana celotna diskovna množica (disk set)

3) Zagon OS, zagon linuxa

Kratek postopek zagona:

Power-on-self-test, Bootsrapt loader, boot loader program (LILO, GRUB), kernel initialization, init program.

Linux področja:

kernel initialization:

Preveri sistemske naprave, Identifikacija specifičnih naprav, Jedro (Kernel) - Zagotavlja, da bo strojna oprema delala to, kar hočejo programi. Funkcije: Preskus bistvenih naprav (CPE, konzola, pomnilnik), Preskus strojnih podsistemov (I/O vodila, omrežni vmesniki, trdi diski, CD-ROM pogoni, disketni pogoni, pomnilne naprave), Inicializacija datotečnega sistema, Spreminjanje konfiguracije jedra

Init program:

init() začne življenje kot nit jedra in konča kot proces na uporabniškem nivoju (/sbin/init), Init je predhodnik vseh procesov (vendar brezposeln) "seje" otroke, Teče v uporabniškem načinu (user mode) (do jedra dostopa preko sistemskih klicev). Funkcije: Pregleda datoteko /etc/inittab, preko nje glede na nivo izvajanja požene skripte, ki se nahajajo v datotekah v imeniku /etc/rc.d (servisi za beleženje sistemskih obvestil, vzpostavitev mreže), požene procese, ki omogočijo prijavo na sistem na (tekstovnih terminalih, lahko tudi grafični uporabniški vmesnik).

4. V jeziku C imamo naslednji stavek: FILE *kazalec Kaj je kazalec? Kaj pomeni FILE?

FILE oz. drugače file pointer. Omenjeni tip je defeniran v stdio.h knjižnici. FILE si lahko predstavljamo kot strukturo, ki drži podatke o datoteki. Moramo pa uporabiti FILE *, saj določene funkcije spremenjajo te informacije in moramo zato podatke o datoteki podajati po referenci. V našem primeru je kazalec spremenljivka,

ki kaže na določeno mesto v pomnilniku kjer so shranjeni podatki o določeni datoteki (ko enkrat datoteko odpremo z fopen).

5. Opiši značilnosti troplastne arhitekture pri middleware. Kaj so prednosti, kaj slabosti?

Primer troplastne arhitektura je poslovna aplikacija, kjer imamo Predstavitevna logika (1 plast), njena funkcija je pretvorba rezultatov in nalog v razumljivo obliko za uporabnika. Druga plast se imenuje procesna logika, ki skrbi za premik oz. prenos podatkov med 1 in 3 plastjo. Njena funkcija je odločanje, računanje. Zadnja plast se imenuje pomnilna logika kjer shranjujemo in pridobivamo podatke, ki jih nato pošljemo nazaj k uporabniku.

Prednosti so: Skalabilnost, Tehnološka fleksibilnost, Dolgoročno cenejša rešitev, Boljše prilagajanje sistema poslovnim potrebam, Zmanjšanje tveganja. **Slabosti:** Visoki stroški v začetku, Orodja in usposabljanje, Izkušnje, Nekompatibilni standardi, Pomanjkanje kompatibilnih orodij za končnega uporabnika.

6. Razloži, v čem se razlikuje klic oddaljene procedure (RPC) od klica navadne funkcije

Glavna razlika RPC-ja od navadne funkcije je ta, da omogoča klice funkcij na oddaljenem računalniku asinhrono ali sinhrono. Lahko bi rekli, da RPC simulira lokalni klic funkcije, ki se zgodi pri navadni funkcije, le da gre pri RPC dejansko pokliče procedure na oddaljenem računalniku. Več o RPC pa je odgovorjeno v zgornjih vprašanjih.

7. Kakšne vloge nudi računalnik, na katerem teče WinServ2008

Vloge so sledeče: Domenski strežnik, Datotečni strežnik (File server), DNS strežnik, Aplikativni strežnik (Application server), Terminal (Terminal server), Print server.

8. Katere vsebovalnike vsebuje AD?

Vsebovalniki - Hierarhični elementi, urejeni v drevesno strukturo. Vsebovalniki v aktivnem imeniku vključujejo:

- Gozdove (Forests)
- Drevesa (Trees)
- Domene (Domains)
- Organizacijske enote (Organizational units)
- Položaje (Sites)

9. Povej kaj več o nalagalniku v BIOS-u.

BIOS nam služi kot bootstrap loader pri zagonu računalnika (kaj je to je napisano pri odgovoru za LILO zgoraj). "Bootstrap loader" locira jedro (operacijskega sistema), ga naloži v pomnilnik in sproži njegovo izvajanje. BIOS je shranjen v ROM-u (se ne izbriše vsebina ob prekenitvi napajanja).

Teorija:

Divjak:

raid, spooler, buffer overflow, active pages, dns porazdeljeni sistemi, nfs, member member off, buffer overflow, inštalacija OS-a, vprasanja iz prosojnic! (kljuci, ntfs,stripped value, konfiguracija op.sistema)

Jager:

komunikacije med uporabnikom, procesi,datotecni sistem, vsebina tabele-i vozlišca, procesi (fork,zombi,sirota), signali, zacetni procesi, kakšne spremenljivke ma bash (globalne in lokalne), aktivni znaki v bashu,ponavljalne strukture v bashu (for, while, until...predvsem so važne razlike med while in until), ukaz expr...

10. Naštej razlike pri delu med sistemskimi klici in knjižnicami?

Sistemski klici:

Zahteva jedro

Nizek nivo

Prenos podatkov

Ni konverzije

Neformatiran tok

Datotečne številke

Počasnejše

Knjižnice:

Ni nujno, da je dana zahteva jedru

Visok nivo

Prenos blokov

Je konverzija

Formatiran tok
Kazalci(FILE)
Hitrejše

Buffer overflow:

Do tega pride, ko skuša proces shraniti podatke v buffer, ki so daljši od fiksnega bufferja beyond the boundaries of a fixed-length [buffer](#)). Povzroča nepravilno delovanje, nepričakovane napake, izvajanje nepoblašene kode itd.

NFS:

Je Network File System protokol, ki je bil prvotno narejen s strani SUN-a. Uporabniki omogoča, da na svojem računalniku dostopa do datotek preko omrežja na omrežnih napravah na tak način kot dostopa do svojih lokalnih diskov.

NTFS:

NTFS je standarden datotečni sistem vključen v Windows NT. NTFS je naslednik datotečnega sistema FAT. Ima boljšo podporo meta podatkom, uporablja bolj kompleksne podatkovne strukture za večjo hitrost, zanesljivost in zasedanje prostora. Dodatno so izboljšali še podporo za varnost ter access control list (ACL).

Spooler oz. spooling:

Spooling je proces prenašanja podatkov na tak način, da jih shranimo na začasno mesto do katerega nato dostopa nek drugi program, ki bo potreboval te podatke pri kasnejši obdelavi/procesiranju. Primer spooling procesa je moč najti pri napravah, ki pišejo hitreje od naprave, ki bere te podatke. S spooler procesom napravi, ki bere omogočimo dostop do podatkov ne glede na hitrost, ki jo ima naprava, ki piše podatke.

Active Server Pages (ASP)

Je prvi server-side skriptni jezik za generiranje dinamičnih spletnih strani. Najprej je bil dodan kot dodatek za ISS 4.0, kasneje pa je bil na voljo kot zastonjska komponenta pri Windows server 2000.

Nekaj dodatka od plonkcev:

UNIX: Op. sistem je namenjen za več uporabniške sisteme. Podpira delo večih procesov naenkrat. Interakcija s sistemom poteka preko X terminala. Je razvojno usmerjen sistem. Napisan v jeziku C. Sestavljajo ga **JEDRO**(kernel), **LUPINA** (interpreter ukazov), **PROGRAMI**.

Vse na Unix-u je datoteka ali proces.

LASTNOSTI: prenosljivost aplikacij, modularnost, ukazna vrstica in je poln programski jezik.

PREDNOSTI: posli v ozadju, pošta med uporabniki, poljubni čas proženja procesov, linki, zaklepanje datotek, komunikacija med procesi.

JEDRO: Nadzor nad aktivnimi procesi in dodeljevanje spomina, oskrba procesov s podatki, upravniki resursov, ima značilnosti virtualnega stroja (iluzija edinega procesa), je programski vmesnik do sistema, sprejema strojne prekinitvene periferije, procesi dajejo zahteve jedru (sistemski klici), delo z datotekami, delo z virtualnim spominom, mrežni protokoli, nitkanje.

PROCESI: uporabnik mora imeti dovoljenje za delo na sistemu. PROCES pomeni izvajanje nekega programa. SLIKA PROCESA je stanje navideznega računalnika v danem trenutku (vsebine registra, stanje odprtih datotek). Proces je izvajanje te slike in se borijo za resurse.

LUPINA: je UNIX-ov interpreter ukazov. Izbira in izvaja zaporedja ukazov in programirane akcije (spremenljivke, pogojni skoki...) zapisane v ukaznih datotekah.

SISTEMSKI KLIC: je zahteva jedru, da nekaj naredi za uporabnikov program. Le jedro sme imeti direkten dostop do sist. Podatkovnih struktur, diskov in periferije.

UPORABA: delo z datotečnim sistemom, nadzor programskih procesov, medsebojno komuniciranje med procesi.

KNJIŽNICE: Standardne (C), Matematične... V okviru knjižnice so dat. Prirejani **DATOTEČNI KAZALCI**, ki kažejo na notranje strukture, kjer so shranjeni podatki o datoteki. Funkcije knjižnice običajno ne rabijo jedra za izvršitev naloge za uporabnikov program. V okviru knjižnice so datotekam prirejani **datotečni kazalci** ki kažejo na notranje strukture kjer so shranjeni podatki o datoteki.

ELEKTRONSKA POŠTA: Hitrost, cena, prenosljivost, prenos datotek, permanentni naslov.

PROTOKOL: nabor pravil ali dogovorov kako komunicirati oz. razumeti preneseno sporočilo.

INTERNET je mednarodna mreža, ki jo sestavljajo individualni računalniki in manjše rač. mreže. Komunikacija v jeziku TCP/IP. Internet naslov (IP naslov, ime domene)

RCP omogoča kopiranje datotek na IP mreži (znotraj LAN)

FTP je standardni mehanizem za prenos ASCII in binarnih datotek med dvema računalnikoma na internetu.

WWW je sistem iskanja informacij na mednarodni mreži računalnikov. Hypertext je vrsta dokumenta, ki vsebuje linke. HTTP je protokol med računalniki WWW. HTP je jezik za kreiranje strani na WWW. URL predpona protokola, domena, pot, ime datoteke.

ARCHIE je orodje za preiskovanje katerega pod. Baza vsebuje sezname anonimnih FTP naslovov. Gopher je orodje za organiziranje informacij na Internetu v povezane večnivojske menije. USENET je vpliven in razširjen konferenčni sistem.

DATOTEČNI SISTEM: navadne datoteke (-), direktoriji (d), posebne datoteke (periferija) (b,c), cevi (p), linki (l).

ABSOLUTNA POT izhaja iz osn. direktorija do ciljnega. **RELATIVNA** pa iz trenutnega do ciljnega.

< VHOD, > IZHOD, >> APPEND, | tok v nek drug proces (pipe)

SISTEMSKI KLICI	KNJIŽNICE
zahtevajo jedro, znaki, ni konverzije, neformatiran tok, datotečne številke, počasnost, nizek nivo.	ni nujno da zahtevajo jedro, visok nivo, bloki (ne znaki), konverzija, formatiran tok, kazalci (FILE), hitreje.

UNIX operacijski sistem je namenjen za več uporabniške sisteme. Podpira delo večih naenkrat. Interakcija s sistemom teče preko X terminala. Je razvojno usmerjen sistem. Napisan je v jeziku C. **Sestavljajo ga: jedro** (kernel), **lupina** (shell) oz. interpreter ukazov in **programi**. Karkoli na UNIXu je datoteka ali proces. **Lastnosti:** prenosljivost aplikacij, vmesnik do uporabnika temelji na ukazni vrstici in je tudi poln programski jezik, modularnost (datotečni sistemi, delo na mreži). **Druge lastnosti:** posli v ozadju, pošta med uporabniki, proces se lahko sprozi ob poljubnem času, linki, zaklepanje datotek, vhodni/izhoni tok je neodvisen od enote, umesniki podatkovnega toka, komunikacija med procesi. **Jedro:** Nadzor nad aktivnimi procesi in dodeljevanje spomina. Proces daje zahteve jedru (sistemski klici). Oskrba procesov s podatki. Je upravniki resursov in jih razporeja procesom optimalno (blokiranje, CPU, en proces naenkrat v časovnem kvantu 10ms). Zagotavlja značilnost virtualnega stroja (iluzija edinega procesa). Je programski vmesnik do sistema (vsebuje gonilnike naprav). Sprejema strojne prekinitve periferije (globalna obravnava). **Delo z datotekami;** delo z vizualnim spominom; nitke (paralelni proces) imajo nekaj skupnega (procesni prostor) pa ne vsega (nimajo istega sklada); mrežni protokol; delo z navidezni

Sistemski klici so zahtevki programa, da naj sistemsko jedro zanj opravi neko akcijo. Ne smemo zamenjavati sistemskih klicev **UNIX** z običajnimi C-jevske funkcijami (podprogrami). **Veljajo naslednje razlike:**

- V času povezovanja (link time) je koda podprogramov kopirana v datoteko z izvršljivim programom; koda, ki jo kličemo s sistemskim klicem, pa je prisotna v samem jedru.
- V času izvajanja (run time), je koda podprograma izvajana kot del kode, vgrajene v izvorni program. Pri klicu sistemske funkcije pa pride do preklopa procesa v sistemski režim.

Sistemske klice lahko delimo na naslednje kategorije:

- Upravljanje procesov
- Medprocesna komunikacija: Signals, pipes, sockets
- Upravljanje z datotekami
- Upravljanje z datotečnim sistemom
- Nadzor okolja in statusne informacije

Vsak proces pri sistemu UNIX ima naslednje atribute:

- nekaj kode
- nekaj podatkov
- sklad
- enoumno številko procesa (PID)

Vrste operacijskih sistemov

Starejši operacijski sistemi so bili pogosto enoopravilni. Omogočali so sočasno izvajanje le enega programa (bolje procesa). Če smo hoteli izvajati več programov, smo morali med njimi ročno preklapljati.

Večopravilni (multitasking) operacijski sistemi dovoljujejo (navidezno) sočasno izvajanje več programov. V resnici računalnik izmenično dodeljuje posameznim programom (bolje procesom) časovne rezine. Največ časovnih rezin običajno dobiva program (proces), ki deluje v ospredju (foreground). Manj rezin dobivajo programi, ki delujejo v ozadju (background). Najmanj časa je posvečeno programom, ki trenutno (še) ne delajo nič.

Mnogouporabniški sistemi dovoljujejo uporabo istega računalnika, včasih celo istega programa, večim uporabnikom. Taki sistemi pogosto delujejo po principu delitve časa med uporabniki (time sharing)

Večprocesorski sistemi temeljijo na uporabi več procesnih enot (CPE). Te so dodeljevane posameznim programskim procesom. Hitrost izvajanja takih procesov je (lahko) večja (ni pa nujno).

Nekateri sistemi lahko dopuščajo (sočasno) uporabo več operacijskih sistemov z uporabo koncepta navideznih strojev (**virtual machines**).

Na področju osebnih računalnikov je prevladal Microsoftov MS-DOS, oziroma njegov dvojček, **IBM PC-DOS**, ki ga je prav tako razvil Microsoft.

Apostrofi in posebni znaki:

```
$ echo $HOME
/home/franc
$ echo '$HOME'
$HOME
$ echo $HOME
$HOME
$ echo ``$HOME"
/home/franc
```