1.)Kerberos
Kerberos je mrežni protokol za avtentikacijo. Njegova glavna naloga je, da nudi močno avtorizacijo za klient/strežnik aplikacije z uporabo kriptogije (https). Pri kerberos protokolu tako strežnik in klient preverita identiteta drug drugega. Kerberos omogoča varno komunikacijo preko nekreptiranega omrežja. Sistem Windows 2000 in naprej privzeto za avtentikacijo uporabljajo Kerberos sistem. Kerberos za svoje delovanje potrebuje trusted third party imenovan key distribution center. Preprosta razlaga delovanja je, da protokol vključuje uporabo listkov (tickets), ki si jih izmenjujeta klijent, ki zahteva dostop in strežnik oziroma aktivni direktorij, ki zagotavlja dostop.

2.)Intrusion detection system
IDS je aplikacija ali naprava namenjena za detekcijo nepoblaščenih dostopov, manipulacijo/onemogočanje sistema, največkrat preko omrežja, naprimer interneta. Slabost IDS je, da ne more zaznati napade skozi pravilno enkriptirane podatke/toka podatkov. IDS sestavljajo tri poglavitvne komponente in sicer Sensors, Console, Engine. Sensors skrbijo za generiranje varnostih dogodkov, Console za spremljanje dogodkov in nadzor na senzorji, Engine pa skrbi za shranjevanje dogodov, ki jih generirajo senzorji in nato sprožanje alaramov glede na dogodke, ki jih sprejme.

3.)Aktivni znaki
<, >, & -preusmerjevanje, | - pipe, $ -pridobitev vsebine spremenlivke, '', ', \ -citiranje drugih znakov

4.)Vidljivost spremenljivk v programu, funkciji
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. Spremenkjivke v skriptu so globalne. Spremenljivke, ki so lokalne v funkciji, se deklarirjo s predpono local. Te spremenljivke so vidne samo v okviru funkcije, ki jih vsebuje.

5.)Katere ponavljalne strukture poznaš (while, until, for) in napisat strukturo / predvsem so važne razlike med while in until)
while - stavki znotraj se izvajajo dokler je pogoj izpolnjen - while [$COUNTER -lt 10]; do
until - stavki znotraj se izvajajo dokler pogoj ni izpolnjen - until [$COUNTER -lt 10]; do
for – s pomočjo for zanke gremo čez seznam besed v nizu znakov - for i in $(ls); do

**6.)Razlika oz. razlaga ",',´, **
Za izpis posebnih znakov v output (naprimer \"),
Za substitucijo izpisa komande z lastno komando, naprimer **echo `date`**

7.)exec,(razlaga) in kaj je pri tem z vh./izh. tokom
Zamenja sliko procesa z drugim procesom. Vhodni/izhodni tok se ohrani.

8.)shift (vgrajen ukaz za premikanje pozicijskih arg.)
Pri uporabi ukaza shift se pozicijski parameteri premaknejo za določeno število (N) v levo. Pozicijski parametri od N+1 do $# so preimenovani v spremenljivke imenovane od $1 do $#-N+1. Ponavadi se uporablja v while loop zanki, ko nam število parametrov ni vnaprej znano in zato pri vsaki ponovitvi zanke pomaknemo parametre za eno levo dokler ne pridemo do konca.

9.)expr (kaj dela)
Prebere expresion, ga izvede in rezultat vrne na standardni izhod. Podpira tako integerje kot nize.

10.)Kaj naredi eval
Ovrednoti argumente/ukaze, jih združi v en ukaz, ki ga nato izvede, rezultat pa je nato na voljo kot izhodni status funkcije. Primer: y=`eval ps ax | sed -n '/ppp/p' | awk '{ print $1 }'` # Finding the process number of 'ppp'.
kill -9 $y # Killing it

11.)Kaj, če predhodnik nima wait-a
Če se predhodni proces konca pred naslednikom naslednik postane sirota. Postane pa sirota zato, ker predhodnik nima waita in se zato konca pred naslednikom.

12.)Začetk-končanje procesa (sistemski klici, zombiji, sirote)
Pri incializaciji 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 – naslendik
exec() – zamenja sliko procesa z drugim procesom
exit – konča z delom (vrne predhodnem procesu status procesa)
wait - sistemski klic omogoča čakanje predhodnika, da njegov naslednik zaključi z delom
kill – ubije proces (sigkill = exit; Sigstop = stop, sigcont

Če se predhodni proces konca pred naslednikom naslednik postane sirota.
Če naslednik konca pred predhodnikom predhodnik postane zombie.

Če child konča z delom preden je predhodnik izvršil sistemski klic *wait*, se deloma razkroji in se tako okrnjen ohrani; oziroma postane "zombie". Če gres list-at procese s ps vrne rezultat <exiting> al pa <defunct>.

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

14.)Vgrajeni ukazi lupine
file, break [n], continue [n], cd [dir], echo [arg, …], eval cmd …
exec cmd …, exit [n], export [var …], expr expr, let expr
pwd, read var …, return [n], set [arg …], shift [n], sleep n
test args …, trap [cmd] [n] …, unset [var …], wait [n]

Vgrajeni ukaz, kakršen je ,,`read``, ga bo izvedla trenutna lupina, sicer pa bo izveden v podlupini.

15.)Raid
RAID - standard za povezovanje diskov in upravljanja z njimi:
Mirroring (RAID1) - pomembna zanesljivost
Striping (RAID0) - pomembna zmogljivost

16.)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 procesorm napravi, ki bere omogočimo dostop do podatkov ne glede na hitrost, ki jo ima naprava, ki piše podatke.

17.)Active Server Pages (ASP)
Je prvi microsoft 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.

18.)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 nepoblaščene kode itd.

19.)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 omrežih napravah na tak način kot dostopa do svojih lokalnih diskov.

20.)Konfiguracija op.sistema
Kofiguracija kot spreminjanje upo imen, passwordov, odstranjevanje nepotrebnih komponent, dodajanje zaščitnih plasti,...

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

22.)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.
cevi (pipes), ce sta na istem rac, sockets ce v mrezi

23.)Zacetni procesi
Pri incializaciji 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().

24.)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 odaljenem računalniku asinhrono ali sihrono. Lahko bi rekli, da RPC semulira lokalni klic funkcije, ki se zgodi pri navadni funkcije, le da gre pri RPC dejansko pokliče procedure na odaljenem računalniku. Več o RPC pa je odgovorjeno v zgornjih vprašanjih.

25.)Active Directory
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 atribute. 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.

26.)Uvod v sistemske klic
Sistemski klici so zahtevki programa, da naj sistemsko jedro zanj opravi neko akcijo. Ne smemo zamenjavati sistemskih klicev *UNIX* z običajnimi C-jevskimi funkcijami (podprogrami).

V UNIXu vsi uporabniški programi in aplikacije uporabljajo vmesnik sistemskih klicev za dostopanje do sistemskih viro kot so diski, tiskalniki, spomin,... Vmesnik sistemskih klicev v UNIXu ponuja množico sistemskih klicev (C funkcij). Naloga vmesnika sistemskih klicev je zagotoviti celovitos 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 sistemskega 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 edini način za dostop do funkcij jedra kot so datotečni sistem, večopravilni mehanizem, komunikacija med procesi.

27.)Inodes
Inode je podatkovna struktura na UNIX podatkovnih sistemih kot je UFS. Inode shranjuje osnovne podatke o dodatki, imeniku kot so: tip datoteke, dolzina v byte-ih, st referenc, lista blokov ki jih zaseda, pravice dostopa do datoteke, datum zadnjega dostopa, podatki o lastniku in njegovi skupini. Vsaka datoteka ima svojo inode datoteko, ki je povezana preko i-number številke.

28.)Dns porazdeljeni sistemi
Da omogočimo 99.9% zanesljivost delovanja v primeru izpada določenega računalnika, je DNS sistem zasnov tako, da je prisotnih ducat DNS strežnikov za vsako domeno in na vrhu tudi 13 zelo močnih root strežnikov, dodatno pa še kopije teh strežnikov, ki so lociranih na različnih koncih sveta.

29.)Zagon in zaustavitev operacijskega sistema
Kratek postopek zagona:
Power-on-self-test, Bootrsrapt loader, boot loader program (LILO, GRUB), kernel initialization, init program.

Bootstrap:
BIOS nam služi kot bootstrap loader pri zagonu računalnika. “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).
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).

30.)Elektronska pošta in drugi servisi (telnet, ftp, ...)
Omenjeni protokoli so internetni protokoli. Vsi ti protokoli sedijo nad TCP/IP protokolom

FTP – omogoča prenos datotek iz/na oddaljeni strežnik.
Telnet – omogoča prijavo na oddaljeni računalnik, tipkovnica in monitor se obnašata kot, če bi bila dejansko priključena na oddaljeni računalnik, torej izpis na oddaljenem računalniku je viden na našem lokalnem monitorju. Naš vpis pa se dejansko vpiše na oddaljenem računalniku.
Email – omogoča enostavno pošiljanje sporočil, potrebno je le vediti elektronski naslov prejemnika

31.)Skriptno programiranje
Lepilo: lepljenje obstoječih večjih komponent.
Kompleksnost je v povezovanju.
Primeri: GUIs, poslovne aplikacije
Primeri skriptnih jezikov: bash, Tcl, Visual Basic, Perl, JavaScript

Značilnosti:
Vse je predstavljeno na enak način (n.pr. z nizi).
Pomen je določen z uporabo.
Rezultat: lepljenje, enostavna ponovna uporaba.
Primer: spremenljivkeVisual Basic.
Primer: Filtri LINUX : select | grep blabla | wc

Stroga tipizacija otežuje lepljenje in ponovno uporabo:
Tipi, vmesniki omejujejo uporabo.
Potrebna je konverzija kode in ponovno prevajanje.
Binarne aplikacije to otežujejo.

Glavni moto: Manj učinkovita uporaba računalnikov, bolj učinkovita uporaba ljudi.

32.)Ukaz awk, Ga lahko primerjamo z "grep" in "sed"?
Awk lahko na kratko opišemo kot orodje za preprosto procesiranje teksta in kot programski jezik za izvajanje kompleksnih operacij nad dobljenim tekstom.

Grep – grep ukaz išče med datotekami ali standardnim izhodom za vrstice, ki ustrezajo določenemu regularnem expresionu, rezultat nato izpiše v standarni izhod. Primer: grep apple fruitlist.txt

Sed oz. or stream editor. SD je uporabljen za izvajanje osnovih transformacij na tekstu, ki ga preberemo iz datoteko ali stdin. Rezultat je nato poslan na izhodni tok. Editor ne spreminja originalnega inputa, ki ga prebere (torej deluje na kopiji originalnega inputa). Glavna razlika SED editorija od vi/ed je da lahko tekst, ki ga preberemo tudi filtriramo ter, da nam ni potrebno komunicirati z sed, kar ga naredi močno orodje za ponavljajoče se naloge editiranja določene vsebine pri velikih/dolgih dokumentih.

Awk – glavna funkcionalnost awk-ja je iskanje skozi datoteke za vrstice oz. bloke podatkov, ki ustrezajo določenim paternom. Ko dobimo zadetek awk izvede določene akcije/operacije nad tem blokom/vrstico teksta. Glavna razlika awk-ja je, da je awk data-driven, kar pomeni, da mi defeniramo podatke na katerih želimo delati in kaj narediti, ko pridemo do takih podatkov.

33.)Kaj je hipervizor
Hipervizor, poznan tudi kot Virtual machine monitor (VMM), je program za virtualizacijo strojne opreme, ki omogoča delovanje večih opercijskih sistemov na določenem računalniku hkrati.

Imamo dve vrste hipervizorjev. Prvi so native, bare-metal kar pomeni, da se izvajajo direktno na strojni opremi računalnika, nad hipervizorjem pa nato teče operacijski sistem.

Drugi tip hipervizorjev teče kot normalna aplikacija na operacijskem sistemu, kar pomeni, da je hipervizor zadnja/vmesna plast med operacijskim sistemom na katerem teče ter operacijskem sistemu, ki ga virtualiziramo. Torej, operacijski sistem – gost teče na tretjem nivoju, hardware, OS, hipervizor. Primer: VMware Server.

34.)Zakaj se uporablja virtualizacija, prednosti
Virtualizacija je način kako doseči večjo porazdeljenost. Sicer se dejanska moč računalnika na katerem tečejo virtualizacijski sistemi malce zmanjša, zaradi presežla. Virtualizacija je način kako maksimalno izkoristit investicije v strojno opremo.

35.)V lupini bash napisite program zaporedje, ki izracuna in zapise prvih n stevil zaporedja ki je definirano kot Y(i)=Y(i-1) + 3 * Y(i-2), i>=2 in sta Y(0) = 1 in Y(1) = 2. Program naj prebere n kot argument ukazne vrstice.

```
Y[0]=1
Y[1]=2
echo "Y[0] = 1"
echo "Y[1] = 2"
n=$1
i=2
while [ $i -lt $n ]; do
    declare -i a=${Y[`echo "$i-1"|bc`]}
    declare -i b=${Y[`echo "$i-2"|bc`]}
    Y[$i]=`echo "$a + ( 3 * $b )" | bc`
```