

SzámRend – kérdezz-felelek

- 1. Milyen jellemző paraméterei vannak egy mai asztali számítógépnek?**
Processzor, memória, háttértár, alaplap tulajdonságai, hűtés típusa stb.
- 2. Mi a cache szerepe a mikroprocesszorban?**
Cache: gyorsítótár. Különböző órajelű egységek sebességkülönbségeinek kompenzálása.
Pl.: gyorsabb órajelű egység küld adatot lassabbra: az adatok átmenetileg a cache-ben tárolódnak, így a gyorsabb egységnek nem kell a lassabbra várnia.
- 3. Milyen a kettes komplementű számábrázolás?**
Fixpontos számábrázolási módszer. A pozitív számokat hagyományos módon alakítjuk át kettes számrendszerbe, míg a negatívokat a konverzió után bitenként negáljuk, majd hozzáadunk 1-et.
- 4. Mit tud az UTF-8 kódolásról? Mire jó?**
[8-bit Unicode Transformation Format] Változó hosszúságú UNICODE karakterkódolási eljárás. Bármilyen UNICODE karaktert képes reprezentálni, ugyanakkor visszafelé kompatibilis a 7 bites ASCII szabvánnyal.
- 5. Soroljon fel legalább 3 memóriatípust!**
 - RAM
 - ROM
 - Flash
 - Cache
- 6. Soroljon fel olyan hétköznapi eszközöket, amelyekben „számítógép” van!**
Okostelefon, okosóra, légkondicionálók, autók, házimozzi-hifi rendszerek, közlekedési lámpák stb.
- 7. Mi a különbség a szerver és egy kliensgép között?**
Szerver: nagy teljesítményű hardware vagy software, ami a következőket biztosít(hat)ja más számítógépek (kliensgépek) számára egy hálózaton keresztül:
 - tárolt adatok elérése
 - szerver által előállított adatok elérése
 - a szerver hardware erőforrásainak kihasználása (pl. nyomtató, processzor)
 - egyéb szolgáltatások eléréseKliens: hálózaton keresztül kapcsolódik a szervergéphez, igénybe veszi annak szolgáltatásait.
- 8. Soroljon fel legalább 3 operációs rendszert!**
Windows, Unix, Linux (+ disztrók), MacOS, iOS, Android, DOS stb.
- 9. Mi a különbség az SSH és a Telnet kapcsolat között?**
Az SSH kapcsolat biztonságos (titkosított) kommunikációs csatornát épít ki a kliens és a szerver között, titkosított adatátvitellel dolgozik. (A Telnet nem titkosít...)

10. Milyen szolgáltatásai vannak egy mai operációsrendszernek?

Programok futtatása, be-/kivitel, szerkeszthető fájlrendszer, hibakeresés/-kezelés, erőforrás elosztás, védelem, hálózati kapcsolat, kommunikáció, kliens-szerver megoldások, közös és osztott háttértárak, több felhasználói fiók stb.

11. Mit ért Shell alatt? Nevezzen meg legalább kettőt!

Rendszerhéj: felület, amely segítségével a felhasználó kommunikálhat a kernellel (rendszermag). Parancssori: Bash, POSIX shell, PowerShell, Command Prompt. Grafikus: (nem vettük, de pl Windows meg minden, ami nem ~== DOS)

12. Mi az alias? Hol találkozott vele?

Shell parancs: egy szó helyettesítése egy másikkal. Többnyire parancsok helyettesítésére. pl. PowerShell

13. Milyen UNIX fájlrendszer jellemzőket tud megemlíteni?

A fájlrendszer kiindulópontja a „root” („/”); a fájlok és könyvtárak rendszere fát alkot; hierarchikus szerkezetű; kétféle bejegyzés: könyvtár, fájl; eszközök is rendelkeznek fájlnevével; linkelés: fájlra (vagy annak tartalmára) mutató bejegyzés

14. Milyen fájlrendszereket ismer?

FAT32, exFAT, NTFS, ext2, ext3, ext4.

15. Milyen fájlnevé konvenciókat ismer UNIX-ban?

Kisbetűs, ékezet nélküli elnevezések; elválasztásnak „-” vagy „_”; rejtett fájlok: „.”-tal kezdődnek; különleges karakterek használhatóak, de nem javallottak

16. Milyen fájl jellemzőket ismer UNIX-ban?

A fájl a rendszerben egy strukturálatlan bájt-sorozat (nincs fájlvég-jel). A rendszer nyilvántartja a fájl hosszát, utolsó módosítás időpontját, a fájlrendszer hány pontjáról hivatkozunk erre a fájlra stb.

17. Magyarázza el a UNIX-ban lévő alapvető hozzáférési jogosultsági rendszert!

Minden könyvtárbejegyzés vagy fájl (különböző típus), vagy könyvtár. Minden bejegyzés jogosultságai 3 csoportot alkotnak: a tulajdonos/felhasználó jogosultságai (u), a csoport jogosultságai (g), bárki/mindenki jogosultságai (o). Minden csoportnak 3 féle jogosultsága lehet: olvasási (r), írási (w), futtatási (x). Módosítás: chmod parancs.

18. Milyen kiegészítő jogokat ismer UNIX-ban?

setuid: a futtatható állomány a fájl tulajdonosának jogosultságaival fut.

setgid: a futtatható állomány a fájl csoportjának jogosultságaival fut.

sticky bit: a könyvtár fájljai csak a tulajdonos által módosíthatóak.

19. Milyen célt szolgál a UNIX-folyamatok prioritása?

A UNIX rendszer folyamatai felváltva futnak, gyors váltásokkal. A prioritás-rendszer segítségével a rendszer hosszabb futásidőt oszt ki a fontosabb folyamatoknak.

20. Mit tud az idézőjelekről a UNIX rendszerben?

"": stringek deklarálása; különleges karaktereket (pl. változónevek) behelyettesíti.
'': minden karakter saját magát jelenti; nincs behelyettesítés.
`: parancs behelyettesítése.

21. Mit jelent az stdin, stdout?

stdin: standard input; alapértelmezett bemenet: billentyűzet
stdout: standard output; alapértelmezett kimenet: monitor
(stderr: standard error output; alapértelmezett hibakimenet: monitor)

22. Hány szűrő kell egy csővezetékhez?

Kettő, vagy több.

23. Mondjon példát arra, hol használhat reguláris kifejezéseket?

Csővezeték: pl grep parancs, sed parancs stb.

24. Mi az az ASCII kódtábla?

8 (eredetileg 7) bites karakterkódolási tábla, az angol ábécé kis- és nagybetűit, számokat, írásjeleket, és néhány vezérlőkaraktert tartalmaz.

25. Mik azok a környezeti változók?

Olyan globális változók, amelyek befolyásolhatják különböző folyamatok futását (pl. PATH, TEMP)

26. Adja meg, hogy UNIX-ban milyen típusú(ak) lehet(nek) a változó tartalma(k)!

UNIX-ban minden változó szöveges típusú.

27. Mit jelent a parancsbehelyettesítés?

Mikor egy parancs kimenetét egy szövegbe szeretnénk visszahelyettesíteni, akkor a szövegben belül, ``-k közé írjuk a parancsot. Futáskor lefut a parancs is, a végeredmény pedig bekerül a szövegbe (változóba stb.).

28. Sorolja fel, hogy milyen műveletek (aritmetikai, logikai) léteznek UNIX shellben!

Alapvetően csak szövegösszefűzés van közvetlenül a shellben. Más (pl. aritmetikai, logikai) műveletekhez parancsokra van szükség (pl. expr, bc, test).

29. Melyik shell utasításnak van befejezési eredménye?

Mindegyik shell utasításnak van befejezési eredménye: 0 = sikeresen lefutott, 1 = sikertelen.

30. Hogyan implementálják a logikai értékeket a UNIX shellben?

"Test" parancs segítségével (vagy "[]"): lefut-e az adott parancs. Ezért 0 = igaz, 1 = hamis.

31. Hogyan készíthetünk összetett logikai kifejezést UNIX shell scriptben?

"Test" parancs –a vagy –o operátoraival egymás után fűzünk több kiértékelést. A precedenciát módosíthatjuk a "(" és ")" karakterekkel.

32. Lehet-e paramétereket kezelő függvényeket definiálni UNIX alatt?

Igen, C-szerű függvényeket lehet definiálni, amely a paramétereket shell script-hez hasonló módon kezeli; visszatérését a "return" kulcsszóval lehet beállítani.

33. **Tudja-e (és ha igen, hogyan) futtatni a végrehajtási jogosultság nélküli shell scriptet?**
Igen, a script tartalmát át kell adni az adott shellnek, és azt futtatni. (sh -v, bár nem tudom, ez mit csinál)
34. **Mi az IFS?**
[Internal Field Separator] Alapértelmezett elválasztójel a shellben (alapesetben szóköz/tabulátor/soremelés); értékadással megváltoztatható (lehet több karakter).
35. **Milyen feladatokat tud elvégezni a SED-del?**
Komplex behelyettesítések, cserék (első, utolsó, összes elem cseréje; csere regex alapján stb.)
36. **Írja le általánosan egy SED parancs szintakszísát!**
sed [paraméter] [cím] s/minta/új_minta/[jelző]
([] = opcionális)
37. **Mi a különbség a SED használatában a " és a ' idézőjel használata között?**
" ": változók, különleges karakterek behelyettesítésre kerülnek.
' ': nem történik behelyettesítés.
38. **Jellemezze az AWK lehetőségeit!**
C nyelvi lehetőségeket ad; tipikus szűrő, pótolja a shell/sed hiányosságait; soronkénti szövegszerkesztést tesz lehetővé.
39. **Adja meg, hogy milyen parancsblokkok találhatók AWK-ban!**
3 parancsblokk:
Első sor előtti inicializálás: BEGIN { }
Soronkénti feldolgozás: { }
Utolsó utáni inicializálás: END { }
40. **Használható-e az AWK aritmetikai feladatok megoldására?**
Igen, C-ből ismert aritmetikai jelek működnek AWK-ban, ill. van numerikus függvénykészlete.
41. **Mi az MBR és mi a feladata?**
Master Boot Record (~Fő indítórekord): A merevlemez egyik particionált része, ahonnan az operációs rendszer betöltődik. A számítógép bekapcsolása után az alaplap itt keresi, és ide adja át a vezérlést.
42. **Írja le a UNIX-LINUX boot folyamatot!**
6 lépésből áll:
1. Bios
a. Az alapvető be- és kimeneti folyamatokat biztosítja
b. Megkeresi, betölti és lefuttatja a boot betöltő programot (MBR)
2. MBR
a. Lásd: 41. kérdés
3. GRUB
a. Grand Unified Bootloader ~ A főbb bootfolyamatokat ez indítja el
b. Kialakítja a kezdeti RAM-ot
c. Futtatja a Kernel-t

4. Kernel ~ rendszermag
 - a. Kiosztja a feladatoknak, hogy a hardvereket mennyire használhatják.
 - b. ~ multiplexálás
 - c. Átadja a vezérlést az init-nek
 5. Init
 - a. /etc/inittab néven van elmentve
 - b. Eldönti a Linux futásának a szintjét
 - c. 0: megáll, 6: reboot (nem érdemes)
 6. Futásszintű programok
 - a. Átadja a vezérlést a felhasználónak a további műveletekhez
43. **Írjon le legalább egy UNIX-LINUX management lehetőséget!**
File manager: Felhasználói felületet biztosít a felhasználónak a fájl – és mappakezeléshez a hierarchikus rendszerben és az alapvető fájlkezelő műveleteket is elvégzi (másolás, átnevezés std.)
44. **Milyen hálózati kapcsolódási lehetőségeket ismer?**
Fix: kábelalapú, Ideiglenes: WiFi, 3g, 4g, cellás, műholdas
45. **Mit ért csomagkapcsolt hálózat alatt?**
Nincs előre kiépített út a kommunikációra, továbbá a küldeni kívánt adatot csomagokra bontják, és ezek külön-külön kerülnek elküldésre.
46. **Mit ír le az OSI modell?**
[Open Systems Interconnection Reference Model] A számítógépek kommunikációjához szükséges hálózati protokollt határozza meg.
47. **Nevezzen meg hálózati topológiákat!**
Centralizált: Egy központi felületen (szervergépen, buszon) történik a kommunikáció. (csillag, gyűrű, teljes, fa)
Decentralizált: Nincsen meg az a központi felület, amely a kommunikációt gyorsítaná, ezért a kiépítése vagy nem túl hatékony, vagy költséges (Sín, busz)
Hibrid: előző kettő részenkénti alkalmazása (csillag-sín)
48. **Mi a feladata egy switch-nek?**
A hálózat felépítésében a kapcsolás és útválasztás feladatát, valamint a portok összekapcsolását a switch, mint hálózati eszköz valósítja meg.
49. **Mi a feladata a routernek?**
Hálózatok összekapcsolása és az ezeken történő adatfolyamatok irányítása, lebonyolítása.
50. **Hogyan jellemezné az IPv4 címeket?**
IP címek 32 bites szerinti felírása, bájtanként felírva egy 0 és 255 közti egész számmal, ponttal elválasztva.
51. **Hol található a DNS-sel az informatikában?**
[Domain Name System] Számítógépes hálózatoknál, pl. internet
52. **Mi az a DHCP?**
[Dynamic Host Configuration Protocol] Dinamikus állomáskonfigurációs protokoll, ez osztja ki a lokális IP-t, hogy ne kelljen manuálisan konfigurálni.

53. Milyen szerver elérési módozatokat ismer?

SSH, TELNET, HTTP, FTP.

54. Mire szolgál a HTTP protokoll?

Állományok feltöltése, letöltése, főként HTML dokumentumokat.

55. Mi történik, ha a public_html könyvtárban nincs index.html fájl?

Kilistázza a mappát és onnan lehet letölteni a tartalmát, vagy ha az privát, akkor hibát jelez a böngésző.

56. Hogyan lehet jelszóval védeni egy weben lévő könyvtárat?

.htaccess és .htpasswd fájlal.

57. Mit ért virtuális host alatt?

Egy webcímre más néven hivatkozunk.

58. Mit jelent az SSI vagy CGI jog a webszervereken?

A webszerver dinamikus interfészeket (SSI, CGI) futtathat, az azokhoz szükséges paramétereket (pl. stdio, környezeti változók) elérheti.

59. Milyen Windows script lehetőségeket ismer? Van egyáltalán?

Igen: Batch, WSH [Windows Scripting Host], PowerShell

60. Mi biztosítja PowerShell-ben az „autoexec.bat” szerepét?

gpedit.msc (ötletem sincs, hogy kéne értelmezni a kérdést, ennyit leltem a diák közt)

61. Hogyan biztosítják PowerShell alatt a biztonságos script futtatást?

ExecutionPolicy segítségével, amely alapértelmezetten “Restricted” - azaz nem engedélyez futtatást. Átállítás: Set-ExecutionPolicy –ExecutionPolicy UnRestricted. Lehetséges policy értékek: Allsigned, Remotesigned, Bypass.

62. Milyen a PowerShell parancsok felépítése?

Ige-[Modul]Főnév alakúak a parancsok. (pl: Get-Help)

Ige: milyen műveletet hajt végre (pl: Get, Set, New, Remove)

Modul: megmondja milyen modulban található a parancs. Nem minden parancs tartalmaz modulnevet. Nem mindenhol szükséges.

Főnév: Milyen adaton vagy dolgon hajt végre valamit a parancs.

63. Soroljon fel PowerShell-ben legalább két változóláthatósági formát!

- global: mindenhol látható az egész PS-ben
- local: függvény vagy szűrő hatókörében lesz látható. Miután a függvény befejezte a futást a benne lévő változók elvesznek.
- script: a teljes scriptben látható lesz
- private: Olyan lokális változó amihez a „gyerek” környezetek nem férnek hozzá.

Egy környezetben definiált változókat, a környezetében használhatjuk, az ebből származó függvény, script látja. Azonos nevű esetén a lokálisat látjuk alaptól. Változót definiálhatunk a scope-jával együtt: \$[scope:]név

64. Hogyan irányítjuk át PowerShell-ben az output-ot? Lehet?

Új fájl, felülírás: „Hello” > szoveg.txt
Hozzáfűzés: „szia” >> szoveg.txt
Nincs < vagy << fajta átirányítás!

65. Hol és mire használható a dot sourcing?

Függvény, változó szintjének a módosítása. Mivel függvényen belül is definiálható függvény és az nem hívható közvetlenül.
pl: . .\UtilityFunctions.ps1 futtatás után az UtilityFunctions script függvényei és változói használhatóvá válnak a jelenlegi scopeban is.

66. Mit jelent a PowerShell függvények nevesített paraméterezési lehetősége?

Van lehetőségünk elnevezni egy script paramétereit, és így nem az \$args változóba kerülnek.
param(\$x, \$y)
“A ` \$x={0}” -f \$x
“A ` \$y={0}” -f \$y

67. Mi a különbség a mikroprocesszor és a mikrokontroller között?

A mikrokontroller általában egyetlen lapka ami tartalmaz egy processzor magot, adatmemóriát, programot és programozható ki/bementei perifériákat. Általában valamilyen konkrét feladatra optimalizált cél-számítógép.

A mikroprocesszor (CPU), néha kiegészítve memória vezérlővel. Bináris jeleket fogad és dolgoz fel. Más alkatrészekkel együtt szokták használni. Általános felhasználású.

68. Mi a „Harvard architektúra” legfontosabb jellemzője?

Egy számítógép-felépítési elv, amelyben a programkód és az adatok külön, fizikailag elkülönített útvonalakon közlekednek a processzor felé.

69. Milyen operációs rendszerben lehet 128 bites egész számot definiálni?

64 bites operációs rendszerben.

70. Mire használható a lebegőpontos számábrázolás?

Lehetővé teszi a valós számok tárolását és kezelését, véges tárhely esetében.

71. Mit jelent az aszimmetrikus kódolás?

Az aszimmetrikus kódolás azt jelenti, hogy az adatok kódolásához és dekódolásához két külön kulcsra van szükség egy helyett. Ez a két kulcs a publikus és a privát kulcs: a publikussal titkosítunk, és a priváttal lehet dekódolni (ehhez szükséges a publikus is).

72. Hány számot takar az RSA publikus vagy privát kulcsa?

Mind a kettő 2 számot takar (egy prímszámot – hatványkitevő a képletben, ez különbözik kulcsenként – és egy másikat, amire nézve azonos maradékosztályba kerül a kódolt adat – ez azonos a privát és publikus kulcsnál)

73. Mit jelent a bináris FTP lehetősége?

Az üzenetküldő minden fájl bájtonként küld el a vevőnek, a vevő a bájtfolyamot tárolja.

74. Mit jelent a szöveges FTP? Létezik egyáltalán?

Igen, szöveges fájlok küldésére alkalmas. Az adatot szükség esetén átalakítják 8 bites ASCII karakterkódra, a vevő fordított módszerrel dekódol.

75. **Hogyan irányíthatja át a szabványos bemenetet PowerShellben?**
Sehogy: nincs input átirányítás (<, <<) PowerShell-ben.
76. **Mi helyettesíti a "here input" funkciót PowerShellben?**
Read-Host –Prompt (nem vagyok biztos ebben, a „here input”-ról semmit nem találtam)
77. **Mire használható a profile.ps1 állomány? Van a UNIX-ban megfelelője?**
Globális PowerShell profil létrehozására, gyakran használt változókat, alias-okat és függvényeket lehet vele definiálni. UNIX megfelelője: \$HOME/.profile
78. **Mit értünk PowerShell modulon?**
Hasznos függvények, alias-ok, változók definíciójának gyűjtőhelyét.
79. **Elég-e a core PowerShell modul a registry módosításához? Miért?**
Igen, mivel a Core adatforrásai közé tartozik a Registry is.
80. **Hogyan használhatja PowerShellben a parancsbehelyettesítést?**
A „\$(parancs)” szintakszissal.
81. **Hogyan készíthet ciklust SED scriptben?**
Nincs ciklus SED scriptben
82. **Jellemzően milyen állományokat talál az /etc könyvtárban?**
Konfigurációs állományokat és a rendszer számára fontos adatbázisokat tartalmaz.
83. **Mire szolgál a hálózati csomagok TTL adata?**
A TTL [Time To Live] adat az adott csomag élettartama. Ha eléri a 0-t, a csomag törlődik. Olyan adatcsomagoknál hasznos, amelyek elkeverednek, nem jutnak el a vevőhöz.
84. **Mit mutat meg a "Netmask"?**
A LAN (alhálózat) méretét.
85. **Mit értünk "nem routolható" IP címen?**
Erre az IP-re a router nem tud adatot küldeni, pl. privát hálózatok.
86. **Mire szolgál a "gateway"?**
Ez az IP egy kivezető út, pl. /sbin/route -n.
87. **Mi az "ARPANET" és milyen lehetőségeket teremtett?**
[Advanced Research Projects Agency Network] A '60-as évek projektje. Csomagkapcsolt hálózat. Megteremtette: TCP/IP, FTP, Mail, NVP (Hangtovábbítás, kezdetleges).
88. **Mi az IPv6? Miért van rá szükségünk?**
128 bites (8 db 16 bites szám) címet használó IP cím. Az egyre több internetre kapcsol készülék miatt az IPv4 modell előreláthatóan nem lesz elég, hogy az összeset megcímezze.
89. **Mondjon példát a "setuid" bit hasznosságára!**
Egy parancs a tulajdonosának jogosultságaival fusson.

- 90 **Mire jó a "sticky bit"?**
UNIX-ban könyvtárak jellemzője: könyvtárban csak saját fájlt törölhetők.
- 91 **Milyen célt szolgál az ACL használata Unix-Linux rendszerben?**
Hogy a hagyományos jogosultságrendszeren túl más jogokat is tudjunk rendelkezésre bocsátani
- 92 **Létezik Windows rendszerben az ACL lehetősége?**
Igen.
- 93 **Mire használhatjuk a setfacl vagy getfacl parancsokat Linux rendszer alatt?**
Kiterjesztett jogosultságrendszerbeli (ACL) jogosultságokat állít be (setfacl) és ír ki (getfacl)
- 94 **Minden fájlrendszerben hasonló módon (pl setfacl) lehet ACL jogokat állítani?**
Igen
- 95 **Mi az analóg- digitális jelek közti alapvető különbség?**
Az digitális jelek kvantáltak, időben és értékben diszkrét (meghatározott értékeket vesznek fel). Az analóg jelek nem kvantáltak, tetszőleges értéket felvehetnek tetszőleges időben.
- 96 **Mi az adat, cím, vezérlő sín feladata?**
Címsín: eszközök címezését szolgálja, szélessége 32 vagy 64 bit.
Adatsín: a processzor adatokat küld vagy fogad ezen keresztül. 32 vagy 64 bit.
Vezérlő sín: a processzor vezérlőjeleket küld vagy fogad ezen keresztül. (min. 10-15)
- 97 **Hogyan készíthetünk szűrőt UNIX illetve Powershell alatt? Lehet?**
Igen, egy parancs után "|" -t írunk, majd ezt szűrőparancsok (grep, cut stb.) követnek.
- 98 **Mi a lényegi különbség a UNIX ls és a PowerShell Get-ChildItem parancsának eredménye között?**
ls string értéket ad vissza, Get-ChildItem objektumtömböt

ls mappákon és fájlkon dolgozik, Get-ChildItem képes pl. a registryben is navigálni
- 99 **Unix vagy Windows PS környezetben tud használni reguláris kifejezést?**
Unix shellekben a különböző parancsok támogatják, pl grep, sed

PowerShellben a string osztály támogatja a -split, -match és -replace metódusokban
- 100 **Milyen speciális jelentése van annak, ha Unix rendszerben egy fájlnevet .-tal kezdődik?**
A fájl rejtettnek minősül, ls nem jeleníti alapból, ahogy a fájlkezelők sem.
- 101 **Mikor használhatóak jól a reguláris kifejezésekben létrehozható csoportok és miért?**
Amikor összetett keresésben pl. csak azt tudjuk, hogy 1 db betűt/számot/karaktert keresünk, de nem tudjuk, mi az.
x db karakter esetén: [a-z]{x}, ha betű, [a-zA-Z]{x}, ha kis- vagy nagybetű, [0-9]{x}, ha szám stb.

102 Mi a lényegi különbség a UNIX shell és a PowerShell csővezetéken áthaladó adatok között?

A UNIX shell csővezetékén szöveges adatok haladnak át, a PowerShell-én pedig objektumok.

103 Milyen eszköztárral rendelkezünk UNIX és PowerShell szkriptek írásához?

UNIX script: vi, vim, nano

PowerShell: PowerShell ISE

104 Milyen kiterjesztésűnek kell lennie egy PowerShell és egy shell szkriptnek?

PowerShell: .ps, .ps1

shell script: pl. .sh (nem kötelező kiterjesztés)

105 Lehet paramétere egy szűrőnek? Ha igen, adjon meg egy tetszőleges példát, ha nem, miért nem!

Igen, pl. `cut -f1 -d";"`, vagy `wc -w/-c stb.`

106 Mi a különbség az stdout, stderr csatorna között? Léteznek PowerShellben?

stdout: alapértelmezett kimenet; sikeresen lefutott programok ide írnak ki eredményt.

stderr: alapértelmezett hibakimenet; hibaüzenetek ide kerülnek kiírásra.

Igen, létezik PS-ben (Write-Output – Write-Error).

107 Mi a "probléma" az egyes komplementű számábrázolással?

0-nak van egy pozitív és negatív alakja is: redundáns.

108 Hogy lehet az stdin csatornát átirányítani Powershellben?

PowerShell-ben nincs input átirányítás.

109 Mire jó a SED? Mi a leggyakrabban használt parancsa?

A SED szöveges adatfolyamban komplex behelyettesítésekre, cserékre jó. Leggyakoribb parancsa a `'s/minta/új_minta/'`.

110 Lehet-e egy SED scriptben shell scriptet hívni? Miért?

Nem lehet, mert a SED csak behelyettesítéseket végez. (az indoklás saját interpretáció)

111 Lehet-e shell scriptből SED scriptet hívni? Miért?

Igen, lehet, mert a shell script tud másik scriptet futtatni, és sed parancsokat is végrehajt.

112 Mi dönti el Unix rendszer alatt, hogy a script fájl milyen script?

Ha egy shell szkriptek legelső sora a `#!` (shebang) karaktersorozattal kezdődik, a mögötte álló (teljes útvonalú) értelmező kapja a fájlt futtatásra.

`#!/bin/sh`

`#!/usr/bin/bash`

`#!/usr/bin/python3`

113 Mit értünk az alatt, hogy egy processzor például 10 nanométeres technológiájú?

A nanométer technológiájú nódus egykor a processzor kapuinak, tranzisztorainak méretére utalt.

Manapság marketing nevek a különböző gyártási technológiák között, és a név nem írja le az áramkör komponenseinek tényleges méretét

Kiseb MOSFET méretű processzorok energiahatékonyabbak, és potenciálisan gyorsabbak.

114 Mit jelent, hogy egy processzor 32 vagy 64 bites?

A memória címbusz szélességét írja le.

32 bites processzor 32 bit széles címeket tud lekérdezni, így ~3.5GB memóriát tud megcímezni, míg egy 64 bites ~16EB memóriát.

Minden modern processzor 64 bites, a mobilprocesszorok is.

115 Hogyan tud AWK scriptet készíteni? Tud egyáltalán?

Igen: `#!/usr/bin/env awk -f`

116 Mit értünk IoT eszközökön, lehetőségeken? Lehet ezeket az eszközöket programozni?

[Internet of Things] Internetkapcsolattal rendelkező mikrokontrollerek építhetők a mindennapi használati tárgyakba, így azok távolról vezérelhetők, illetve különböző adatok lekérdezhetők róluk.

117 Mi az AWK program BEGIN és END blokkjának a szerepe?

Mind kettő egyszer fut le: a BEGIN az elején, az END a végén. Emiatt használható arra, hogy pl a BEGIN blokkban kiírjuk a feladat címét, azt elvégezzük, majd az END blokkban kiírjuk az összegzését.

118 Lehet-e karakter kódtáblával adatokat, szöveget titkosítani?

Lehet. Más kódtáblán más karaktert jelöl ugyan az a szám.

119 Hogyan készíthet SED scriptben pl for ciklust?

(:)-al létrehozott címekhez ugrik a vezérlés a b paranccsal, feltétel teljesülésénél ugrik a t paranccsal. Goto-hoz hasonló.

120 Mi a különbség a PowerShell for ciklusa és Foreach-Object utasítása között?

A ForEach-Object parancs végrehajt egy utasítást az adott objektum összes elemén. Az objektumot meg lehet adni csővezetéken keresztül vagy az InputObject paraméterrel.

Ez a dokumentum az ELTE IK Programtervező
Informatikus hallgatói által készített, **nem hivatalos**
segédanyag a Számítógépes Rendszerek c. tárgyhoz.
Használat saját felelősségre!

Készítők:

Ambrus-Dobai Márton, Bahrami Benedek Attila,
Gortka Bence, Hadházi Dávid, Hamrik Szabin,
Csépán Botond

2019 © ELTE IK