

Számrend elméleti ZH

2018.12.11.

ELTE IK Bsc (esti)

Szabó Norbert - YECVA4

1. Milyen jellemző paraméterei vannak egy mai asztali számítógépnek?

Processzor típus (darabszám), memória méret, háttértár, input-output perifériák.

2. Mi a cache szerepe a mikroprocesszorban?

Csökken az adathoz (így a számítás eredményéhez) való hozzáférés költsége (ideje vagy energiája) azzal, hogy fontos adatokat a cache-ben tárolunk. A CPU cache egy kisebb, gyorsabb memória, amihez gyorsan hozzáfér a processzor.

3. Milyen a kettes komplementű számábrázolás?

Egész számok ábrázolására használjuk.

- $-x = \text{negált } x + 1$
- Egy nulla, -128 $+127$ közötti számok egy bájt

4. Mit tud az UTF8 kódolásról? Mire jó?

Karakterkódolás, azaz kódoknak megfeleltetünk karaktereket, így tudjuk ábrázolni. Az UTF-8 kódolásban rengeteg karakter szerepel, az egész világon használható.

5. Soroljon fel legalább 3 memóriatípust!

RAM, ROM, FLASH.

6. Soroljon fel olyan hétköznapi eszközöket, amelyekben "számítógép" található!

Telefon, mosógép, autó, TV, stb.

7. Mi a különbség egy szerver és egy kliensgép között?

Kliens: egy felhasználót szolgál ki

Szerver: több felhasználót (klienst) szolgál ki

8. Soroljon fel legalább 3 operációs rendszert!

iOS, Windows 10, Android, Linux

9. Mi a különbség az ssh és a telnet kapcsolat között?

Az ssh titkosított, míg a telnet nem.

10. Milyen szolgáltatásai vannak egy mai operációs rendszernek?

- Kliens-szerver különbségek
- Közös osztott háttértár használata
- Közös nyomtatási szolgáltatás
- Szervizek kezelése
- Felhasználók kezelése

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

Klasszikus felhasználói felület programja.

Windowsban CMD.

Unixban pl SH, BASH.

12. Mi az az alias? Hol találkozott vele?

Az alias-szal parancsok alternatíváit hozhatjuk létre, pl rövidítésnek.

Például "ls -l" indokolt aliasa lehet az, hogy "dir".

Találkozhatunk vele bármilyen UNIX környezetben, pl pandora.inf.elte.hu

13. Milyen UNIX fájlrendszer jellemzőket tud megemlíteni?

- Hierarchikus szerkezet
- Csak kétféle bejegyzés: könyvtár vagy fájl (az eszközök és linkek is fájlnevet kapnak)
- Az egész naplózott, ami nagyobb biztonságot ad

14. Milyen fájlrendszereket ismer?

Ext2, Ext3, NTFS, FAT32, stb.

15. Milyen fájlnev konvenciókat ismer a UNIX-ban?

Korlátlan hosszú név, tetszőleges karakter.

Konvenció szerint lehetőleg ne használjunk ékezetet, szóközt és speciális karaktereket.

Nincs kötelező fájlkiteresztés.

Ha a fájlnev ponttal kezdődik, akkor takart állomány.

16. Milyen fájl jellemzőket ismer a UNIX-ban?

Név, méret, létrehozás dátuma, tulajdonos, tulajdonos csoportja, hard link szám, jogosítványok

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

3x3-as rendszer.

Három kategóriának adhatunk jogot:

1. Tulajdonos
2. Csoport
3. Mindenki más

Az alábbi három jogot lehet nekik kiosztani:

1. Olvasás
2. Írás
3. Végrehajtás (futtatás)

18. Milyen kiegészítő jogokat ismer a UNIX-ban?

Pl. parancs a fájl jogaival fut (nem a futtató jogaival), vagy parancs a fájl csoport jogaival fut.

19. Milyen célt szolgál a UNIX-ban a folyamatok prioritása?

Az a folyamat kapjon hamarabb/több processzoridőt, amelyik fontosabb.

20. Mit tud az idézőjelekről a Unix rendszerben?

Módosítják a klasszikus értelmezést.

21. Mit jelent az stdin, stdout?

stdin: billentyűzet, alapértelmezett bemenet

stdout: monitor, alapértelmezett kimenet

22. Hány szűrő kell egy csővezetékhez?

Legalább 2.

23. Mondjon példát arra, hol használhat reguláris kifejezéseket?

Egy szövegminta általános megadására való, például ha csak azokat a sorokat akarjuk kiírni egy txt fájlból, ami szóközzel kezdődik.

24. Mi az ASCII kódtábla?

Egy jelkészlet, amivel kódokhoz rendelünk karaktereket.

25. Mik azok a környezeti változók?

Az operációs rendszer változói fontos adatokkal, a programon kívül van, de a programnak elérhető, pl PATH.

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

UNIX-ban minden változó valójában szöveg; amikor dolgozunk vele, akkor értelmezhetjük pl. számként is.

27. Mit jelent a parancsbehelyettesítés?

A 'parancs' parancsot végrehajtja, az eredmény kerül a helyére. Pl:
ki_vagyok='whoami'

28. Sorolja fel, hogy milyen műveletek léteznek a UNIX shellben?

Egyetlen művelet: szövegösszefűzés (shellben minden változó szöveg), közvetlenül semmilyen más művelet nem támogatott.

29. Melyik shell utasításnak van befejezési eredménye?

Mindegyiknek van eredménye, ez jelzi a sikerességét vagy sikertelenségét.

30. Hogyan implementálják a logikai értékeket a UNIX shellben?

1 igaz, 0 hamis.

31. Lehet-e paramétereket kezelő függvényeket definiálni a UNIX alatt?

Klasszikusan nem, viszont a függvényt argumentumokkal is meg lehet hívni, amit az kezelni tud.

32. Tudja-e (és ha igen, hogyan) futtatni a végrehajtási jogosultság nélküli shell szkriptet?

Igen, ha így indítom:
Sh scriptneve.sh

33. Mi az az IFS?

Az alapértelmezett elválasztó karakter (Internal Field Separator).

34. Milyen feladatokat tud elvégezni a sed-del?

Komplex behelyettesítéseket, soronként értelmezve.

35. Írja le általánosan egy sed parancs szintakszísát!

sed [par] [cím] s /minta/új_minta/[jelző]

36. Mi a különbség a sed használatában a " és a ' idézőjel használata között?

```
x=Tibi;  
echo Laci ügyes!|sed "s/Laci/$x/" ===> Tibi ügyes!  
echo Laci ügyes!|sed 's/Laci/$x/' ===> $x ügyes!
```

37. Jellemezze az awk lehetőségeit!

Shell hiányosságait pótolja szövegfeldolgozáshoz, C nyelvi lehetőségek, tipikus szűrő.

38. Adja meg hogy milyen parancsblokkok találhatók az awk-ban!

Végrehajtás előtte parancsok => soronkénti végrehajtás ciklus => végrehajtás utáni parancsok

39. Használható-e az awk aritmetikai feladatok megoldására?

Igen, sőt, egyéb matematikai műveletek is vannak benne.

40. Mi az MBR és mi a feladata?

Master Boot Record, ezt töltjük be először a memóriába, ami továbbadja a vezérlést arra a partícióra, ami ténylegesen bootolható.

41. Írja le a UNIX-LINUX boot folyamatot!

Boot sector (BIOS) => Kernel => Init processz

42. Írjon le legalább egy UNIX-LINUX management lehetőséget!

Például kézi módosítása az /etc/hosts fájlban.

43. Milyen hálózati kapcsolódási lehetőségeket ismer?

Wifi, Bluetooth, [Soros](#), stb

44. Mit ért csomagkapcsolt hálózat alatt?

Csomagküldés- és fogadás alapú hálózati kapcsolatot. Az adatokat csomagokra bontjuk, és azokat továbbítjuk.

45. Mit ír le az OSI modell?

7 rétegre bontja a rendszerek felépítését, mindegyiknek egymásra épülő saját feladata van.

46. Nevezzen meg hálózati topológiákat!

Csillag, fa, gyűrű.

47. Mi a feladata egy switch-nek?

Hálózati összekapcsolás.

48. Mi a feladata a routernek?

Lokális hálózatok összekapcsolása globális hálózatokkal.

49. Hogyan jellemezné az IPv4 címeket?

4 bájtos címek. 2 része van: hálózati cím + számítógép azonosító.

50. Hol találkozik a DNS-sel az informatikában?

Domain Name Service, domaint kapcsol IP-hez, például böngészés közben találkozunk vele.

51. Mi az a DHCP?

Dynamic Host Configuration Protocol: hostok konfigurálására való, lehet automatikus vagy kézi.

52. Milyen szerver elérési módokat ismer?

Terminálos elérés, webes elérés.

53. Mire szolgál a HTTP protokoll?

HyperText Transfer Protocol, hálózati protokoll, amivel fel- és letölteni tudunk.

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

Megkapunk egy tartalomjegyzéket: fájl- és mappalistát (ha ez nincs kifejezetten tiltva a szerveren).

55. Hogyan lehet jelszóval védeni egy weben lévő könyvtárat?

.htaccess és .htpasswd fájlal.

56. Mit ért virtuális host alatt?

Virtuális host használata során más néven hivatkozunk egy címre.

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

Könyvtárra vonatkozik, futtathatunk CGI és SSI függőségeket.

58. Milyen Windows szkript lehetőségeket ismer? Van egyáltalán?

Van, például Batch, PowerShell.

59. Mi biztosítja PowerShell-ben az "autoexec.bat" szerepét?

Belépéskor automatikusan lefutó parancsokat tartalmaz, ma már Group Policyket használunk (GPEDIT)

60. Hogyan biztosítják PowerShell alatt a biztonságos szkript futtatást?

Az ExecutionPolicy állítgatási lehetősége, ami alapból Restricted.

61. Milyen a PowerShell parancsok felépítése?

Ige-Főnév [paraméterek]

62. Soroljon fel PowerShellben legalább két változó láthatósági formát?

global, local

63. Hogyan irányítjuk át PowerShellben az outputot? Lehet?

Például > vagy >> jelekkel.

64. Hol és mire használható a dot sourcing?

Függvényen belüli függvénydefiniálás, például arra jó, hogy rendszerbe szervezzük a függvényeinket.

65. Mit jelent a PowerShell függvények nevesített paraméterezési lehetősége?

Nem hagyományos paraméterezés; egyből nevet is adunk a paraméternek (változónak).

66. Mi a különbség a mikroprocesszor és mikrokontroller között?

Mikroprocesszor csak önmagában a processzor, ami a számításokat végzi, nincs más eleme. A mikrokontroller egy számítógép, egy vezérlőegység, aminek része a processzor, emellett lehet saját RAM-ja, stb.

67. Mi a "Harvard architektúra" legfontosabb jellemzője?

A programkód és az adatok külön, fizikailag elkülönített úton jutnak el a processzorhoz.

68. Mire használható a lebegőpontos számábrázolás?

Tört számok ábrázolására.

69. Mit jelent az asszimetrikus kódolás?

Legelterjedtebb formája az RSA; a célja, hogy a nyilvánosan közlekedő (pl interneten áthaladó) adatokat titkosítsuk.

70. Mit jelent a bináris ftp lehetősége?

Pontosan átmásoljuk a fájlokat byte-ról byte-ra.

71. Mit jelent a szöveges ftp? Létezik egyáltalán?

A plain text alapú fájlknál hasznos, mert egyből konvertál fájlrendszer szerint (a sortörések másképp vannak Unix és Windows rendszerben).

72. Hogyan irányíthatja át a szabványos bemenetet PowerShellben?

Sehogy, erre alkalmazhatunk trükköket, pl először beolvasunk valamit, és azt irányítjuk át.

73. Mi helyettesíti a "here input" funkciót PowerShellben?

Read-Host

74. Mire használható a profile.ps1 állomány? Van a UNIX-ban megfelelője?

Felhasználóhoz kötött aliasokat és függvényeket tartalmaz. UNIX megfelelője .profile

75. Mit értünk PowerShell modulon?

Funkciógyűjtemény, összefüggő scriptgyűjtemény, amit használhatunk a scriptünkhöz.

76. Elég-e a core PowerShell modul a registry módosításához? Miért?

Igen, mert a registry bejegyzések ugyanúgy működnek mint a mappák/fájlok.

77. Hogyan használhatja PowerShellben a parancsbehelyettesítést?

Ugyanúgy, mint bash-ben:
valtozoneve=`parancs`

78. Hogyan készíthet ciklust sed scriptben?

A sed alapvetően ciklikusan működik: például egy szövegfájl minden sora a ciklus egy lefutását vonja maga után.

79. Jellemzően milyen állományokat talál az /etc könyvtárban?

Rendszerkonfigurációs fájlokat.

80. Mire szolgál a hálózati csomagok TTL adata?

Time to live, azaz meddig érvényes a csomag. TTL nélkül a csomagok örökre keringenének a hálózaton.

81. Mit mutat meg a "Netmask"?

Azt, hogy egy adott alhálózaton, az elejétől fogva, hány bitnek kell megegyeznie az IP címekben.

82. Mit értünk "nem routolható" IP címen?

Olyan IP cím, ami nem lehet kapcsolatban a világhálóval.

83. Mire szolgál a "gateway"?

Eszköz, ami két hálózat közötti átjárást teszi lehetővé.

84. Mi az "ARPANET" és milyen lehetőségeket teremtett?

Amerikai kutatóintézetek közt kiépített széleskörű csomag-kapcsolt hálózat, ebből lett a TCP/IP.

85. Mi az IPv6? Miért van rá szükségünk?

IP címeket takarnak az IPv6 nevek. Az IPv4 címek elfogynak, egyszerűen nagyobb tartományra van szükségünk, amit használhatunk.

86. Mondjon példát a "setuid" bit hasznosságára!

A felhasználó tudja módosítani a saját jelszavát.

87. Milyen célt szolgál az ACL használata Unix-Linux rendszerben?

Access Control List, azt tartalmazza ki és milyen joggal fér hozzá a fájlhoz.

88. Létezik Windows rendszerben az ACL lehetősége?

Igen.

89. Mi az analóg- digitális jelek közti alapvető különbség?

Az analóg jelek folyamatosak, például hangok.

A digitális jeleket nem folytonosak, adatcsomagokra bontjuk, például egy kép.

90. Mi az adat, cím, vezérlő sín feladata?

Címsín: eszközök címezése

Adatsín: adatküldés

Vezérlőszín: vezérlőjelek küldése

91. Hogyan készit szűrőt UNIX illetve Powershell alatt? Lehet?

UNIX-ban és PowerShell-ben a parancs mögé írjuk a szűrőt | jellel elválasztva, így a parancs kimenete lesz a szűrőhöz a bemenet, és annak kimenete a leszűrt eredmény.

92. Mi a lényegi különbség a UNIX ls és a PowerShell Get-ChildItem parancsának eredménye között?

A UNIX ls csak felsorolja a mappa tartalmát, a PowerShell Get-ChildItem pedig részletesebb információkat ad. (UNIX ls -l => PowerShell Get-ChildItem)

93. Melyik környezetben tud használni reguláris kifejezést a UNIX, a Windows PowerShell világában vagy mindkettőben?

Mindkettőben.

94. Milyen speciális jelentése van annak, ha Unix rendszerben egy fájlnev .-tal kezdődik?

Rejtett fájl.

95. Mikor használhatóak jól a reguláris kifejezésekben létrehozható csoportok és miért?

Amikor az eredmény, amit elvárunk az nem a teljes kifejezés, hanem annak csak egy része. Például csoportokra bontjuk a kifejezést, és kiválasztjuk, hogy melyik csoportra van szükségünk, így nem kell további regexet futtatni az első regex futtatás eredményén.

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

UNIX: szöveg

PowerShell: objektum

97. Milyen eszköztárral rendelkezünk UNIX és PowerShell szkriptek írásához?

Bármilyen szerkesztő, például Joe, Notepad++, PowerShell Editor...

98. Milyen kiterjesztésűnek kell lennie egy PowerShell és egy shell szkriptnek? Van egyáltalán előírás vagy szabadon megválasztható?

Shell esetében a kiterjesztés lényegtelen, PowerShell esetében például .ps1

99. Lehet paramétere egy szűrőnek? Ha igen, adjon meg egy tetszőleges példát, ha nem, magyarázza meg, hogy miért nem lehetséges!

Igen, például a grep szűrőnek:
cat asd | grep -i "tehen"

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

Az stdout a standard kimenet, az stderr a hibakimenet. PowerShellben Write-Host és Write-Error.

101. Mi a "probléma" az egyes komplementű számábrázolással?

A nulla kétféleképpen ábrázolható, és a műveletek nehézkesek.

102. Hogy lehet az stdin csatornát átirányítani Powershellben?

Sehogy, erre alkalmazhatunk trükköket, pl először beolvasunk valamit, és azt irányítjuk át.

103. Mire jó a SED? Mi a leggyakrabban használt parancsa?

Szerkesztési műveleteket végez. Leggyakrabban használt parancs az "s", ami szövegcserét jelent.

104. Lehet-e egy SED scriptben shell scriptet hívni? Miért?

Nem, mert nem erre találták ki; shell scriptben lehet SED scriptet hívni.

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

A script első sorában lehet definiálni.

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

Egy nyomtatott áramkör technológiát jelöl, de már nincs köze a fizikai méretekhez.

107. Mit jelent, hogy egy processzor 32 vagy 64 bites?

Másképp kezeli a memóriát: 32 bit vagy 64 bit alapon. A 64 bites architektúra a jobb, ha nagyobb memóriaigényünk van.

108. Hogyan tud AWK scriptet készíteni? Tud egyáltalán?

Tudok, a script első sorában jelzem, hogy ez AWK script.

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

IoT = Internet of Things, a tárgyak internete, például okoshűtő, stb. Lehet programozni, kell programozni hogy elérjük a kellő funkcionalitást, mint fejlesztő. Rendszerint a felhasználók nem programozhatják ezeket.

110. Mi az AWK program BEGIN és END blokkjának a szerepe?

BEGIN az AWK lényegi (ciklikus) blokkja előtt fut, az END pedig utána, mind a kettő egyszer fut le.