8. Diagnosztika és programozás

1. Jellemezze a „diagnosztika“ fogalmát, hozzon fel példákat az ipar különböző területeiről.

A diagnosztika egy számítógépes hardver eszköz vagy szoftver tesztelésének módszere annak biztosítására, hogy megfelelően működjön. Ezt a tesztelést főként a piacra hozatal előtt szükséges elvégezni. Az után már nem kötelező a diagnosztizálás, viszont egy évben legalább egyszer ajánlott diagnosztikát elvégezni a gép helyes működésének vizsgálatára. (szép leírás, viszont az első mondatot én átfogalmaznám, mivel a kérdés nem konkrétan a számítógépdiagnosztikára vonatkozott, hanem úgy általánosan kérdezte, mi az. Ha azt átfogalmazod, a többi már mehet így utána)

Diagnosztikát legfőképp a számítógépes rendszereknél, illetve a műszaki berendezéseken szokás alkalmazni az iparban. (jöhetne pár konkrétabb példa az iparból, innen meríthetsz ötletet: <https://hu.wikipedia.org/wiki/Diagnosztika> )

* 1. Hajtson végre szoftver diagnosztikát a PC HW helyes működésének vizsgálatára.

A legegyszerűbben használt szoftveres diagnosztikai eszközeink bele vannak építve a BIOS-ba (UEFI), vagy illetve az operációs rendszerben. BIOS-ban használt beépített diagnosztikai eszköze lehetőséget ad megvizsgálni a számítógép összes alkatrészét, mint például a processzort, a memóriát, a tárhelyeinket a képernyő helyes működését és sok mást. (megemlíthetnéd a POST-ot is itt)

Ezen kívül használhatunk operációs rendszerbe beépített alkalmazásokat, mint pl a Windows teljesítményfigyelője vagy a Windows memória diagnosztizálója.

Ha ezek után továbbra is vizsgálataink sikertelenek használhatunk harmadik féltől származó szoftvert. Itt már megtalálható specifikus programok kifinomítva egyes komponensekre, bár nem szabad elfelejteni, hogy ebből sok szoftverért fizetni vagy licencet kell venni. (jó ez, de jöhetne pár példa képekkel, pl. Wi-Fi diagnosztika, program elindításának diagnosztikája – kompatibilitási mód bekapcsolásának módja, eszközkezelő bemutatása – driverfrissítés, csatolhatsz egy képet harmadik féltől származó programról is).

1. A PC hálózatok problémáinak diagnosztikája
   1. Egy Wireshark programban elkapott http-csomagon magyarázza el és írja le az OSI és TCP/IP referencia modellt és jellemezze egyes rétegeit, protokolljait, adategységeit és a hálózatokban történő adatok enkapszulációját (beágyazás)/dekapszulációját (kicsomagolás).

OSI modell:

1 Réteg: A fizikai réteg protokollok leírják a fizikai kapcsolat aktiválására, fenntartására és kikapcsolására szolgáló mechanikus, elektromos, funkcionális és eljárási eszközöket a bitátvitelhez.

2 Réteg: Az adatkapcsolati réteg protokolljai leírják az adatkeretek eszközök közötti cseréjét.

3 Réteg: A hálózati réteg szolgáltatásokat nyújt az egyes adatok darabos hálózaton keresztüli cseréjéhez.

4 Réteg: A szállítási réteg meghatározza az adatok szegmentálására, átadására és újra szerelésére szolgáló szolgáltatásokat a végberendezések közötti egyedi kommunikációhoz.

5 Réteg: A munkamenet-réteg szolgáltatásokat nyújt a bemutató réteg számára a párbeszéd megszervezéséhez és az adatcsere kezeléséhez.

6 réteg: A bemutató réteg biztosítja az alkalmazásréteg-szolgáltatások között átvitt adatok közös ábrázolását.

7 réteg: Az alkalmazásréteg a folyamatok közötti kommunikációhoz használt protokollokat tartalmazza.

TCP/IP modell:

1. réteg - Hálózati hozzáférés: A hardvereszközöket és az adathordozókat vezérli.

2. réteg - Internet: Meghatározza a legjobb utat a hálózaton keresztül.

3.réteg - szállítás: Támogatja az eszközök közötti kommunikációt különböző hálózatokon keresztül.

4. réteg - Alkalmazás: Az adatokat képviseli a felhasználó számára.

Enkapszuláció: utal az adatok összevonására vagy az objektum egyes alkotóelemeihez való közvetlen hozzáférés korlátozására. A beágyazást egy strukturált adatobjektum értékeinek vagy állapotának elrejtésére használják egy osztályon belül, megakadályozva az ügyfelek közvetlen hozzáférését hozzájuk oly módon, hogy rejtett megvalósítási részletek tárhatók fel, vagy megsérthessék a módszerek által fenntartott állapot variánst.

Dekapszuláció: Olyan kapszulázott adatok megnyitásának folyamata, amelyeket általában csomagok formájában küldenek egy kommunikációs hálózaton keresztül. Szó szerint definiálható a kapszula nyitásának folyamata, amely ebben az esetben kapszulázott vagy becsomagolt adatokra utal.

(Jó ez a részletes leírás, viszont hiányzik a Wiresharkos példa, írd le a Wiresharkban a packetelfogás folyamatát, mutass be egy csomagot, ahogy a kérdés kéri – youtubeon biztos van róla valamilyen jó videó kiscreenshotolhatsz pár képet róla, bőven megfelel, de a legjobb egy saját példa beszúrása volna)

* 1. Az OSI modell szemszögéből írja le a hálózatokban előforduló hibák azonosítását és megszüntetését.

broadcast storm – Ez akkor jön létre amikor túlságosan sok switch található a domainben, és ezzel lassítja a kommunikációt: Megoldás: a domain lekicsinyítése, vlanok használata

nincs internet elérés – sokféle képpen történhet ez a probléma, például amikor a kábelok kilazulnak vagy elszakadhatnak. Megoldás: A kábelok megigazítása, kábel kicserélése

(kicsit bővebben jobb lenne, a lényeg, hogy 20 percet lehessen beszélni az egészről, legjobb volna ilyen 5-6 hiba pár mondatban kifejtve).

* 1. Írja le az IPv4 és IPv6 protokollok beállítását a PC hálózati kártyáján. Demonstrálja a diagnosztikai lépéseket annak helyes működésének ellenőrzésére.

Windows 10 operációs rendszerben jobb klikkelünk a internet ikonra, ezzel megnyitva a gépházat. A gépházban kiválasszuk az adapterbeállítások módosítását majdan a megnyílt ablakon kiválasszuk az általunk használt adaptert. Jobb klikkel kiválasszuk a tulajdonságokat és abban az alakban a legördülő képen megkeressük a TCP/IP protokoll 4 illetve 6-os verzióját. Ezt dupla klikkelve megnyílik az ablakunk, ahol beállíthassuk az ip címünket statikus, illetve DHCP-re. Kezdő, illetve tudatlan felhasználóknak ajánlatos DHCP-n hagyni ezt.

(jó leírás, viszont jöhetnének róla képek – ipv4-ipv6-ra is – , nézd meg pl. a Flóri jegyzetét, ő nagyon jól megoldotta az ilyen „lépésről-lépésre” feladatok leírását, abból meríthetsz ötleteket, hogy mennyi kép kell, stb.)

Egy statikus vagy DHCP beállításnak tesztelésére a legegyszerűbb módszer figyelni a internet ikonunkat. Ha az nem jelez hibát nagy valószínőséggel van internet elérésünk. Ezen kívül letesztelhetjük a kapcsolódásunk különféle harmadik fél programokkal vagy a command promp segítségével.

* 1. Írja le az ICMP protokollt és mutassa be annak felhasználását. Mutassa be szemléletesen a diagnosztikánál használt parancsokat: ping, tracert, nslookup, arp, netstat, route print CLI-ben, mutassa be azok variánsait és magyarázza el azok kimenetét.

(jöhetnének a parancsok használatáról screenshotok és magyarázat, hogy hogy kell őket beállítani)

ping – a kapcsolatok tesztelésére használt kommand

tracert – egy bizonyos kapcsolatvonal megfigyelése, ami leírja számunkra mennyi „ugrásba”, illetve milyen routereken keresztül jutott el a célpontba. Itt megfigyelhető akár a packet elvesztését is, ha nincs következő ugráspontunk.

nslookup – e kommand segítségével lefordíthassuk domaineket ip címekre

arp – megmutassa számunkra a számítógépünk által ismert eszközöket az adás tartományban

netstat – Megjeleníti a hálózati kapcsolatokat az Átviteli vezérlés protokollhoz

route - az IP-útválasztási táblázat megtekintésére és kezelésére szolgál

1. Magyarázza el az MS-DOS rendszer működését az OS-ben. Demonstrálja annak alapvető parancsait az OS parancssorban.

Lemezes operációs rendszer, amely az első 8086, 8088… -PC XT, AT processzoros IBM kompatibilis 16 bites személyi számítógépeken használtak elsőként. (jöhetne ide egy kiadási év, amúgy ez az egész pont jó)

Alapja – OS PC-DOS, amelyet az IBM adott el PC-ihez.

További OS-ek: DR-DOS – Digital Research, MS DOS – Microsoft, Novell DOS, Open DOS, Free DOS stb.

Jellemzése:

- Egyfeladatos

- Egyfelhasználós

- Operatív memória alapmemóriája – 640kB

- FAT16 fájlrendszer

- Lemezpartíció maximális mérete 512MB

- Max. 4 hajlékonylemez meghajtó

Az IBM-kompatibilis PC-n az MS-DOS volt a legnépszerűbb.

Alapvető parancsok:

cd – könyvtár váltás

dir - információkat jelenít meg a fájlokról és könyvtárakról, valamint arról, hogy mennyi szabad lemezterület áll rendelkezésre. Alapértelmezés szerint megjeleníti az aktuális könyvtár minden fájljának nevét, méretét és utolsó módosítási idejét.

copy – egy vagy több fájl másolása

del – egy fájl törlése

edit - lehetővé teszi a számítógépen lévő bármely szöveges fájl megtekintését, létrehozását vagy módosítását.

move – egy fájl átvitele másik könyvtárba

ren – egy fájl átnevezése

deltree - fájlok és könyvtárak végleges törléséhez használják a számítógépről (rekurzív törlés)

cls - lehetővé teszi a felhasználók számára, hogy töröljék a képernyőn megjelenő összes tartalmat, és csak egy üzenetet hagyjanak

format – Egy meghajtó adatainak törlése és annak a fájlformátumjának megváltoztatása

1. Magyarázza meg a változó és a konstans fogalmát a programozásban.

Változó – ez egy általunk megadott névvel ellátott adatforma, amit a program futása közben bármikor lekérdezhetünk és megváltoztathatjuk az értékét.

Konstans – ez egy általunk megadott névvel ellátott adatforma, melynek a program futása közben az értéke mindig ugyanaz marad. Ennek értékének változtatása csakis a kódnak inicializálásában lehetséges.

* 1. Jellemezze a kivételeket Java-ban. Mutassa be használatukat.
  2. Jellemezze és mutassa be a listák (list) használatát Python-ban. (jó, talán kép formátumban jobb lenne, ha bevágnád a programkódot, mert úgy a több szín miatt jobban átlátni)

A listákat több elem egyetlen változóban történő tárolására használják. A listák a Python 4 beépített adattípusának egyike, amelyet az adatgyűjtések tárolására használnak.

A listák szögletes zárójelek segítségével készülnek:

thislist = ["apple", "banana", "cherry"]  
print(thislist)

adat beszúrása a listbe:

list.append(item)

adat törlése a listből:

list.remove(element)

list.pop() – az utolsó elemet törli

a list nagysága visszadja a listánkban lévő adatok teljes darabját:

list.Length()

1. Egyértelműen jellemezze a PIC mikroszámítógépekre szánt Assembler nyelvet – a kód formátuma, néhány alapvető utasítás, a magasabb szintű nyelvekkel szembeni különbözőség, a programon belüli hibakeresés alapelve MPLAB X IDE-ben. (Az utolsó 3 pontot nem néztem, mert azt mondtad, hogy dolgozol rajta)

Assembler nyelv:

alacsony szintű, a programozónak ismernie kell a számítógép részeit részletesen, hogy tudjon vele dolgozni. 

Különbözőség:

Mivel alacsony szintű programozás, így a programozónak magát a hardwert is be kell konfigurálnia. Tehát figyelnie kell a regiszterek idejére, ürítésére, szabad memóriára stb.

Hibakeresés:

A legeslegegyszerűbb hibakeresés a szerkesztő tanácsa. Ezzel bármilyen szintaxis hibát megmutat nekünk, és akár segít is annak helyes formázására.

Következő hiba a logikai hibák, például a regiszter helytelen beállítása, itt nem kapunk a szerkesztől tanácsot, hanem itt saját magunk logikájával kell megtalálni a hibát.

1. Magyarázza meg az USART kommunikációt, annak gyakorlati felhasználását és a mikrokontroller munkáját ezzel a kommunikációval (röviden írja le a mikroC nyelv funkcióját UART modulra).

Universal Synchronous/Asynchronous Receiver/Transmitter - egy soros interfész eszköz típusa, amely programozható aszinkron vagy szinkron kommunikációra.

Az USART működése szorosan összefügg a különféle protokollokkal-

A szinkron módú USART-k keretben továbbítják az adatokat. Szinkron üzemmódban a karaktereket időben kell megadni a keret elkészültéig; ha a vezérlő processzor ezt nem teszi meg, akkor ez egy "aláfutási hiba", és a keret továbbítása megszakad.

A szinkron eszközként működő USART-ok karakter-orientált vagy bit-orientált módot használtak. Karakter (STR és BSC) módban az eszköz bizonyos karakterekre támaszkodott a kerethatárok meghatározásához; bit (HDLC és SDLC) módokban a korábbi eszközök fizikai réteg jelekre támaszkodtak, míg a későbbi eszközök átvették a bit minták fizikai réteg felismerését.

A szinkron vonal soha nem hallgat; amikor a modem továbbít, az adatok áramlanak. Amikor a fizikai réteg azt jelzi, hogy a modem aktív, az USART folyamatos kitöltést küld karakterekből vagy bitekből, az eszköznek és a protokollnak megfelelően.

1. Magyarázza el a regiszterek működését és rajzolja meg a következők egyikének bekötési rajzát (memória regiszter, léptető regiszter, körregiszter, Johnson számláló).

A regiszterek gyors írható – olvasható munkatárak. A különböző regisztereket szigorúan meghatározott feladatokhoz vannak hozzárendelve, emiatt korlátozott számú funkció betöltésére alkalmasak. Része az ALU-nak vagy CU-nak. vagy önálló tömb. gy regiszter általában egyszerre csak 1 gépi szó (word – rövid karakterlánc, 1-2 szó általában 2-4 bájt) tárolására alkalmas.

memória regiszter:

léptető regiszter: (shift register)

A shift regiszterek egyik általános felhasználási területe a soros-párhuzamos átalakítás. Erre sok esetben szükség van, mert a jelek soros átvitele egyszerűbb, majd az átvitel után az eredeti byte forma kerül visszaállításra. A shift regiszterekkel kapacitás és induktivitás nélküli késleltető áramkörök alakíthatók ki.

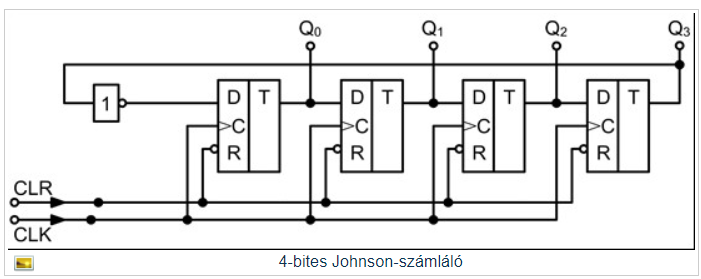
További felhasználása a bitekben való léptetés, azaz csusztatás.

körregiszter:

Egy körkörösen visszacsatolt léptető regiszterből készíthetünk. A léptető regiszterben az adatok az egyik vagy másik (balra vagy jobbra), vagy akár mindkét irányban eltolhatók, léptethetők. Ha a léptető regiszter kimenetét valamilyen módon a bemenetre kötik, akkor visszacsatolás jön létre és a működés ideje alatt a regiszterben egy megadott információ kering. Ezt nevezik körregiszter számlálónak.

Johnson számláló:

A kezdeti állapot beállításához mindegyik tároló kimenetét nullázni kell, így a számláló bekapcsoláskor a főciklusba kerül.Az állapotsorozat jellegzetessége, hogy a számláló a visszacsatoló ágba kapcsolt inverter miatt balról jobbra logikai 1-ekkel, majd 0 bitekkel töltődik fel



A felépítése egyszerű, kombinációs áramköri elemként csak egy invertert tartalmaz, így a működési sebessége is nagy.

n számú flip-flop esetén az 1 és 0 értékű bitekkel való feltöltődés n-n db különböző állapotot eredményez. A megkülönböztethető állapotok száma, vagyis az áramkör modulusa (m) kétszerese a flip-flopok számának (n):

A feladat megoldásához használhat, komponenseket, képeket, ábrákat, laboratóriumi munkáit, számítógép hardvert és szoftvert.