17. tétel:

Új munkaidő-nyilvántartó számítógépes rendszert terveztek a munkahelyén. Ön, mint gazdasági informatikus a programozókkal együtt dolgozik, segítik egymás munkáját.

- **Ismertesse, milyen operációs rendszer szükséges a program telepítéséhez? Csoportosítsa az operációs rendszereket!**
- **Sorolja fel a rendszerelemzési és - tervezési módszereket!**
- **Milyen programozási nyelvet alkalmazna? Ismertesse a programozási nyelveket és jellemzőiket!**
- **Milyen módszertanokat használ a tervezéshez, illetve milyen tervezési szoftverrel valósítaná meg a dokumentációt?**
- **Sorolja fel a munkavégzés nyilvántartásának szabályait!**
- **A programozókkal együtt az Ön feladata az új szoftver átadása. Milyen módszert javasol a rendszer bevezetésére, milyen dokumentumokat kell átadnia a felhasználóknak?**

Operációs rendszer

Az operációs rendszer az a szoftver (program) amely nélkülözhetetlen a számítógép működéséhez. Feladatai közé tartozik többek között a **hardver kezelése**, a **programok futtatása**, és **kapcsolattartás a felhasználó és a számítógép között**.

Az operációs rendszer típusai:

I. ÁLTALÁNOS CÉLÚ OPERÁCIÓS RENDSZER

1. Egyfelhasználós (monouser)

- Egyfeladatos – az első operációs rendszer: a DOS.
- Többfeladatos (multitasking) – egy felhasználó több feladatának időbeni párhuzamos végrehajtása egy gépen: Windows.

2. Többfelhasználós

- Egyfeladatos (batch processing) – kötegelt feldolgozás, mely esetén az operációs rendszer egymástól független munkákat, parancskötegeket (batch) hajt végre egymás után.
- Többfeladatos (multiprogramming) – a CPU-t ütemezési stratégia szerint ciklikusan rendelik hozzá a tárban elhelyezett programokhoz, kihasználva azok várakozási időit.

II. SPECIÁLIS OPERÁCIÓS RENDSZER

1. Hálózati – hálózati funkciók kiszolgálása, pl. Novell NetWare. Az operációs rendszer funkcióit az egymással összekapcsolt hardvereszközök között szét lehet osztani.

2. Valós idejű (real time) – folyamatvezérlési feladatokra fejlesztették ki (gyártási folyamatok, közlekedési lámpák). A gépbe az adatok az érzékelőkről érkeznek. Az operációs rendszer feladata ezek elemzése, és az eredményektől függő vezérlő mechanizmusok aktivizálása.

A **tervezés** során az inkremens alapú technikát javaslom használni.
Az inkremenseket hangolni kell, az egyedi igényekre pedig kiegészítő modulokat kell tervezni.

Inkremens alapú tervezés előnyei:

- A megrendelőnek nem kell megvárnia a teljes rendszer elkészültét, hanem már az első inkremens átadása után használhatja a rendszer legfontosabb szolgáltatásait.
- A korábban kifejlesztett inkremensek tekinthetők prototípusnak, a használatuk során szerzett tapasztalatok beépíthetőek a fejlesztés folyamatába.
- Csökkenti a kockázatot. Az egyes inkremensekben ugyan lehetnek hibák, azonban nem valószínű, hogy ezek az egész projekt kudarcát okozzák.
- A magas prioritású inkremenseket szállítjuk le először, így azok lesznek a legjobban tesztelve.

A programozási nyelvek típusai

Gépi kód

A gépi kód valójában nem nyelv, mivel az a gép számára közvetlenül értelmezhető adatsort jelenti. A gépi kódhoz legközelebb álló nyelv az Assembly nyelv. Ha egy assembly vagy mnemonik nyelv, és az általa előállított gépi kód között egy-az-egy megfeleltetés van, akkor mondhatjuk, hogy az adott nyelv „gépikód szintű".

Fordított nyelvek

A compiler (fordító) gépi kódú programot állít elő, így azt az operációs rendszer már közvetlenül képes futtatni. DOS/Windows architektúrákon az ilyen programok indítása egy COM (command - parancs) vagy EXE (executable – végrehajtható) kiterjesztésű fájl betöltésével kezdődik.

- C programozási nyelvű
- C++ programozási nyelv
- C# programozási nyelv
- Clipper programozási nyelv
- COBOL programozási nyelv
- Pascal programozási nyelv (beleértve az Object Pascal nyelvet)
- Java programozási nyelv

A Java és C# kilóg a sorból, mert a fordítóprogram nem gépi kódra, hanem egy köztes kódra, például Java virtuális gép (JVM) bájtkódjára fordít. Ebből a kódból aztán szükség szerint Java gép JIT-fordítója állít elő gépikódot a program futtatása közben.

Interpretált nyelvek

Az interpreter (értelmező) viszont nem állít elő gépi kódot, a beírt kód végrehajtása lényegében a kód soronkénti értelmezésével történik. Ebben az esetben a kód futtatásához tehát egy külön futtató környezet szükséges, ami gyakran azonos a fejlesztői környezettel. Egyes interpretált nyelveknél (pl. Perl, Python) a fordító először egy átmeneti, a forrásszövegnél hatékonyabb kódot állít elő, ami a soronkénti értelmezéshez képest hatékonyabb szintaktikai ellenőrzést, kisebb kódméretet és a „hagyományos" interpretált nyelvekhez képest némi sebességnövekedést eredményez.

- Awk programozási nyelv
- BASIC
- ECMAScript
- JavaScript, JScript (olyan ECMAScript implementációk, amik megfelelnek az ECMA szabványnak, de attól eltérő eljárásokat is támogatnak)
- Perl programozási nyelv
- Python programozási nyelv
- PHP programozási nyelv
- VBScript.

A compiler (fordító) gépi kódú programot állít elő, így azt az operációs rendszer már közvetlenül képes futtatni.

A gépi kód valójában nem nyelv, mivel az a gép számára közvetlenül értelmezhető adatsort jelenti. A gépi kódhoz legközelebb álló nyelv az Assembly nyelv.

Az interpreter (értelmező) nem állít elő gépi kódot, a beírt kód végrehajtása lényegében a kód soronkénti értelmezésével történik.

Programtervezési lépések:
A programtervezési eszközök közé sorolhatjuk mindazokat az eszközöket, módszereket, amelyek lehetővé teszik a programok specifikálását és tervezését.
- Az adatleírást, specifikációt segítő eszközök:
  - Matematikai és logikai szimbólumrendszer
    - ♣ Függvények (sorozatok, tulajdonságok definiálására)
    - ♣ Állítások, logikai műveletek, kvantorok (elő és utófeltételek pontos leírására)
  - Típusleíró eszközök
    - ♣ Matematikai jellegű típusleírás (pl. vektor ill. rekord direkt szorzatként történő felírása, pl. R^N: N dimenziós, valós elemű vektor)
    - ♣ Pszeudo-nyelv típusleíró része (hasonló, mint a programozási nyelvek típusdefiníciós része, de nyelvfüggetlen)

Adatmodell-leíró eszközök:
- ♣ Bachman diagram, relációs adatbázis esetén alkalmazzuk, tartalmazza az egyes adattáblák szerkezetét, és az adattáblák közötti kapcsolatokat
- ♣ UML osztálydiagram.

Algoritmusleíró eszközök: pl. folyamatábra, stuktogram, Jackson-diagram, pszeudo-kód (mondatszerű leírás), UML aktivitás-diagram
OOP tervezést segítő eszközök
- OOP rendszertervező eszköz: UML (Egyesített Modellező Nyelv):
- Nagyon sokféle diagramtípust tartalmaz, amelyek a szoftverfejlesztés valamennyi fázisát segítik.
- UML diagramtípusok pl.
  - ♣ Használati eset diagram: a feladatspecifikáció elkészítéséhez.
  - ♣ Osztálydiagram: a tervezéshez, tartalmazza a rendszerben található összes osztályt, és azok kapcsolatait, számosságukat
  - ♣ Objektum-diagram: demonstrálja az egyes objektum-példányokat
- CASE eszközök:
  - ♣ Beépített OOP rendszertervezés (pl. UML)
  - ♣ Beépített kódgenerálás egy, vagy többféle objektum-orientált forrásnyelven.
  - ♣ Automatikus tesztelés és dokumentálás

A CASE lehetőségek olyan eszközök, amelyek célja a rendszerfejlesztési tevékenység elemzési- és tervezési fázisaiban végzett munka hatékonyságának növelése:
- A rendszer elemzésével
- A fejlesztési információk rögzítésével és elemzésével
- A projekt irányításával
- A fejlesztéshez kapcsolódó információk és teendők adminisztrálásával

CASE eszközök használatának előnyei:
- A fejlesztési munka gyorsabb
- A felhasználói igények hatékonyabban elégíthetők ki, kevesebb későbbi módosításra lesz szükség
- Az ily módon fejlesztett rendszerek minősége magasabb színvonalú, így rövidebb a tesztelési- és próbaidő
- A rendszer működésének ellenőrzése gyorsabban elvégezhető

Az adott feladat kapcsán a strukturált (Top-down) elemzési és tervezési **rendszerfejlesztési módszertant** célszerű használni, hiszen a munkaidő nyilvántartás elég jól definiált feladat.
A strukturált módszertanok felhasználják a korábbi eredményeket, a strukturált programozás megjelenését, a projektvezetési módszertan kialakulását és a dokumentáció jelentőségének felismerését.

A strukturált módszertan jellemzői:

- Termékszemlélet: nem csupán végtermék van, hanem minden tevékenység előállít, vagy módosít egy terméket.
- Technika (módszerek=technikák). Az egyes termékeket milyen technikával állítják elő.(pl. adatmodellezés, folyamatmodellezés)
- Elemzés felülről lefelé: alrendszerekre, funkciókra, folyamatokra bontás
- Tervezés alulról felfelé: hierarchikus építkezés alapelemekből, pontos, részletes terv.
- Logikai (mi történik) és fizikai (hogyan történik) vizsgálatok szétválasztása. Logikai: a rendszer működésének belső logikája, viszonylag állandó. Fizikai: a tényleges megvalósítás adott eszközökön.
- Fokozatosság és iteráció

A strukturált fejlesztés technikái:

- folyamatmodellek (pl.folyamatábrák): hangsúly a folyamatokon
- adatmodellek (pl. egyedhalmaz-kapcsolat-attribútum (E-K-A) ábrák, és normalizálás): hangsúly az adatokon, adatszerkezeteken és az adatok közötti kapcsolatokon

Munkaidő nyilvántartás fogalma: a munkaidő nyilvántartás a munkaidőket rögzíti, vagyis a munkaidő kezdetét és a munkaidő végét.

Jelenléti ív: a jelenléteket rögzíti, vagyis az érkezés és a távozás időpontját, vagyis többet, mint a munkaidő.

Beléptető rendszer az érkezés-távozás adatait rögzíti, vagyis egy jelenléti ív, hosszabb tartamokat tart nyilván a tényleges munkavégzésnél. Ha az érkezés – távozás idejét tartják nyilván munkaidő kezdeteként és befejezéseként, akkor a nyilvántartott időnél csak kevesebb lehet a munkaidő, mert nem akkor kezdődik a munkavégzés, amikor munkába érkezik a munkavállaló és nem is akkor ér véget, amikor távozik a munkahelyéről.

A **Munka Törvénykönyvének 134. paragrafusa** (2.bek.) a munkaidő nyilvántartás tartalmára vonatkozóan kimondja, hogy a nyilvántartásból naprakészen megállapíthatónak kell lennie a teljesített rendes és rendkívüli munkaidőnek, valamint a készenlét kezdő és befejező időpontjának is.

87. § (1) **Munkanap:** a naptári nap vagy megszakítás nélküli huszonnégy óra, ha a munkarend alapján a beosztás szerinti napi munkaidő kezdete és befejezése nem azonos naptári napra is beosztható.

A naprakész vezetés törvényi követelménye azt jelenti, hogy a nyilvántartás legalább a nap utolsó órájában 24:00 óráig legyen kitöltve. A munkavállalók munkaidő adatait célszerű az események bekövetkezésekor rögzíteni! Nem lehet előre kitölteni, csak akkor, ha a munkaidő nyilvántartás egyidejűleg a munkaidő beosztás is.

134. § (1) A munkáltató nyilvántartja
a) a rendes és a rendkívüli munkaidő,
b) a készenlét,
c) a szabadság tartamát.

A munkaidő nyilvántartás formai megjelenésére nincs törvényi előírás. A munkáltató joga eldönteni, hogy napi, heti vagy havi munkaidő nyilvántartást vezet-e.

A **rendszer dokumentálása** során meg kell különböztetni irányítási és műszaki dokumentumokat.
Felhasználók részére bocsátandó műszaki dokumentumok a következők:

- Követelményspecifikáció

- Logikai rendszerterv
- Fizikai rendszerterv
- Teszterv és tesztforgatókönyvek
- Tesztelési naplók
- Felhasználói kézikönyv
- Telepítési és üzemeltetési kézikönyv

Rendszer bevezetése: Az új rendszerre való áttérés formái a következők:

- *Direkt*: az új rendszer üzembe állításával a régit kivonják a forgalomból
- *Párhuzamos:* egy bizonyos ideig a két rendszer együtt működik
- *Részleges:* a felhasználók egy csoportja próbálja ki az új rendszert
- *Lépcsőzetes*: először bizonyos funkciók végrehajtását vállalja át az új rendszer, majd fokozatos átállás az egészre.

Jelen példánál a különböző módok közül a párhuzamos bevezetést ajánlom, hiszen kevésbé rejt kockázatot, mint például a direkt bevezetés. Itt az új rendszer egy ideig teszt üzemmódban működik.