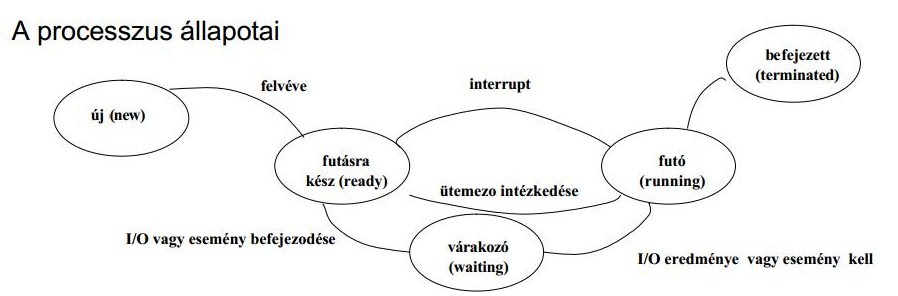
**Elmélet**

1. **Ismertesse és jellemezze a folyamatok 5 különböző állapotát és az egyes állapotok közötti átmenetek lehetőségét.**

processzus állapot (process state) : new, ready, running, waiting, terminated



1. **Röviden jellemezze a First Come First Serve (FCFS) processzor ütemezési stratégiát.**

Sorrendi ütemezés (First-Come First-Served): olyan sorrendben kapják

meg a procok a CPU-t amilyen sorrendben kérik. Nem megszakítható.

1. **Sorolja fel a rendszerhívások 6 kategóriáját és ismertessen 4-et.**

• Processzus kezelés (process management)

• Jelzések (signals), Események (events) kezelése

• Memóriakezelés (memory management)

• Fájl (file), Könyvtár (directory) és Fájlrendszer (file system) kezelés

• Védelem (protection) – adat, user, program, ...

• Üzenetkezelés (message handling)

1. **Ismertesse az operációs rendszerek által a folyamatok reprezentánsának használt Process Controll Block (PCB) fontosabb részeit.**

Processzus - vezérlô blokk (Process Control Block – PCB)

· processzus állapot (process state) : new, ready, running, waiting, halted, sleeping, …

· Program címszámláló (PC) értéke

· CPU regiszterek tartalma

· memória foglalási adatok

· account/user adatok

· I/O státusz információ (a folyamathoz rendelt I/O erôforrások, állományok listája)

1. **Röviden jellemezze a Round Robin processzor ütemezési stratégiát.**

Round Robin: Procok sorba (FIFO) rendezettek; mindig a sorban első,

futáskész kapja a CPU-t, ha elveszti, a sor végére kerül.

Osztályonként lehet egy-egy sor

1. **Soroljon fel és jellemezzen 5 szolgáltatását az operációs rendszereknek.**

· Program végrehajtás (program betöltés és futtatás)

· I/O mûveletek (fizikai szint: blokkolás, pufferezés)

· Fájl-rendszer manipuláció (r, w, c, d)

· Kommunikáció – a folyamatok közötti információ csere (ugyanazon, vagy

különbözô gépeken) Shared memory – Message passing.

· Hiba detektálás (CPU, memória, I/O készülékek, felhasználói programok, ...)

· Nem közvetlenül a felhasználó támogatását, hanem a hatékonyabb

rendszermûködést segítik:

· Erôforrás kiosztás – multiprogramozás, többfelhasználós mûködés

· Accounting – rendszer és felhasználói statisztikák.

· Védelem – minden erôforrás csak az operációs rendszer felügyelete mellett

érhetô el.

1. **Ismertesse 2-3 mondatban azt a 3 leggyakoribb memóriafoglalási stratégiát , amelyek a folyamatok dinamikus memóriájának allokációja által kialakultak a lyukak keresésére.**

· First -fit: foglaljuk le az elsô lyukat, amely elég hosszú!

· Best-fit: foglaljuk le az elég hosszú lyukak közül azt, amelynek hossza legközelebb esik a

szükséges hosszhoz!

· Worst-fit: foglaljuk le a leghosszabb lyukat (ha az elég hosszú)

Sebesség, tárkihasználás: a "first" és "best" jobb, mint a "worst".

· A lyukak adminisztrálása is idôt és memóriát igényel, deadlockhoz is vezethet! (többe

kerülhet mint a szabad hely!)

· 50% szabály: (D. Knuth) a first fit stratégia statisztikai vizsgálata azt eredményezte, hogy n

blokk elhelyezése során n/2 blokk (helye) elvész(!) (ez a kezdeti szabad terület egy

harmada!).

1. **Ismertesse,hogy a memóriamenedzsment esetén a szegmentálás mit jelent és melyek a leggyakoribb elemei amelyek a programozói nézetben megjelennek.**

Szegmensek: egym´ast´ol f¨uggetlen c´ımterek. 2D mem´oria.

• 0-t´ol valamilyen max.-ig c´ımezhet˝o.

• K¨ul¨onb¨oz˝o szegmensek elt´er˝o hossz´uak, m´eret¨uk v´altozhat.

• A c´ımek 2 r´eszb˝ol ´allnak: szegmens sz´ama, szegmensen bel¨uli c´ım.

El˝ony¨ok:

• V´altoz´o m´eret˝u adatszerkezetek k¨onnyen kezelhet˝oek.

• Programok linkel´ese egyszer˝u, ha minden elj´ar´as k¨ul¨on szegmensbe

ker¨ul: ´ujraford´ıt´as eset´en csak a megv´altozott szegmenseket kell

kicser´elni.

• F¨uggv´eny-k¨onyvt´arak k¨onnyen megoszthat´ok proc-ok k¨oz¨ott (pl.

grafikus k¨onyvt´arak csak egyszer vannak a mem´ori´aban).

• K¨ul¨onb¨oz˝o v´edelmi szintek alak´ıthat´oak ki.

H´atr´any: a programoz´onak tudnia kell, hogy ezt a technik´at haszn´alja.

1. **Az operációs rendszerek szemszögéből hogy írná le a védelem szerepét az általános irányelvet és általános megvalósulását? UNIX esetén milyen megoldásokat ismersz?**
2. **Ismertesse egy hagyományos mágneses merevlemez esetén hogy mit jelent a szektor, sáv és cilinder fogalom és mit tud az író-olvasó fejről.**

szektor: egy lemezoldal tortaszelet-szerű osztásai

sáv: egy körgyűrű egy adott lemezoldalon

cilinder: egymás alatt elhelyezkedő sávok együttese

blokk: a legkisebb átvihető adatmennyiség

1. **Miért van szükség a merevlemezek esetén ütemezési stratégiára? Nevezzen meg és egy-két mondatban jellemezzen 3-at. pld:FCFS,SST,SCAN,stb.**

A lemezmûvelet mindig valamelyik sáv valamelyik szektorára történô pozicionálással

kezdôdik:

· keresési idô (seek time): az író/olvasó fej pozicionálása a sávra

· várakozási idô (latency time): várakozás, amíg a kérdéses szektor az író/olvasó fej

elé fordul

· átviteli idô (transfer time): az adat írása / kiolvasása elektronikusan

· A keresési idô nagyságrendekkel lehet nagyobb, mint a másik kettô, ezért a diszk

ütemezés célja a keresési idô minimalizálása.

· A keresési idô arányos a keresési távolsággal (jó közelítéssel).

FCFS (First Come First Served)

· Egyszerûen implementálható, de nyilván nem szolgáltat jó átlagos kiszolgálási idôt.

· Az író/olvasó fej ide-oda "ugrál".

· A fej által összességében megtett "út" hosszú.

· Minden igény elôbb-utóbb kielégül.

SSTF (Shortest Seek Time First )

· Hasonlít az SJF processzus ütemezési stratégiára (az ott optimális volt, ellenben ez itt)

· Nem optimális: az 53, 37, 14, 65, 67, 98, 122, 124, 183 sorrend rövidebb összkiszolgálási idôt

eredményez.

· Az FCFS-nél azért kedvezôbb.

· Éhezéshez (starvation) vezethet!

SCAN (pásztázás)

· Elevátor algoritmusnak is nevezik.

· Ha a fejpozicionálási igények egyenletesen oszlanak el, akkor mindig amikor a fejmozgás irány

a legszélsô cilindernél megfordul, a másik végén van a "tumultus". ? egyenetlen várakozási

idô

1. **Mi az operációs rendszer feladata és milyen szolgáltatásai vannak?**

Feladatok:

• A felhasználó kényelmének, védelmének biztosítása = Kiterjesztett

(Virtuális) gép

• Az emberi felhasználó számára kényelmes kezelési funkciók;

jogosultságok kezelése

• Programok számára futás közben használható eljárások, függvények; a

futó program memóriaterületének védelme

• A rendszer hatékonyságának, teljesítményének maximalizálása =

Erőforrás-kezelés

• Hardver erőforrások: CPU (processzor), memória, I/O

(Bemeneti/Kimeneti) eszközök, fájlkezelés, ... stb.

• Szoftver erőforrások: eljárások, bufferek (átmeneti tárolók),üzenetek,

szolgáltatások, ... stb.Szolgáltatások:

· Program végrehajtás (program betöltés és futtatás)

· I/O mûveletek (fizikai szint: blokkolás, pufferezés)

· Fájl-rendszer manipuláció (r, w, c, d)

· Kommunikáció – a folyamatok közötti információ csere (ugyanazon, vagy

különbözô gépeken) Shared memory – Message passing.

· Hiba detektálás (CPU, memória, I/O készülékek, felhasználói programok, ...)

· Nem közvetlenül a felhasználó támogatását, hanem a hatékonyabb

rendszermûködést segítik:

· Erôforrás kiosztás – multiprogramozás, többfelhasználós mûködés

· Accounting – rendszer és felhasználói statisztikák.

· Védelem – minden erôforrás csak az operációs rendszer felügyelete mellett

érhetô el.