

(463) - 14-25; (R. 323.

www.it.lme.hu / Kondor

FR. 3.

információ: az a szakmai ág ami az

- információval
- információval megteremtés műveletével
- információ feldolgozó rendszerrel

info 1/2.



szakmai

rendszer műveleti-vel foglalkozik

info 1: HW; OP. RENDSZ

- info 2: - hálózatok
- adatbázisok
- programok

- szerep:
- használhatóvá teszi a gépet
 - a fittatás, programok számára támogatást nyújt; eszközök felületét tisztítja
 - ebbe beépítve hatékony felhasználás (kiszerve: tiszta)
 - erőforrás-gazdálkodás

1950 körül az a gépkezelési fogalomkör

azóta HW és szoftver } számítógép megfordultak SW és hálózat

először: HW helyes felhasználás

ma: használhatóvá tétel a felhasználóval

140-as info: kernel + utility program = szerver

kernel & utility
 kernel: számítógép
 nagy, kernel program nem tudnak futtatni, nagy hibák vannak a programokban

SW - szerver.

apendras: modulárisális ball

- pl.: - kancal
- fordilo
- stb. ...

} slóvra ar mind "beindt
varva" ar appendras

ma: ijoa equa tilis lolaq bekkind ar
appendrasle

- pl.: - modisplayer
- liöngörö
- stb. ...

} pl.: ma naqin a
Tista.

völ: virste
pörlötrivöl!

Appendrasle töötänök:

- kanci vendrasch
- eqqram" mantv
- offline - spooling
- multiprogrammáris
- kóta: ... virste ... oo

jállenrák:

- (1) CPU-kibarmilás = $\frac{\text{harmas idó}}{\text{összes idó}}$ (-100%)
- allosalmárisi program
ütántárisit hája eqra
- (2) átkeseti képerny:
- teljénvény: throsegypit
benchnaile programonag
mésóramag futisálisál
rámaljakt
job/h adóte
ar si eqpége

Kanci vendrasch:

- 1750-es Endk
- IBM 4034 zéplis 29 dallár 5év 65 dallár/h amárisális (Bör: 1 dallár/h)
- Open shop vendrasch
pl.: lióramfészene
paraméterek: - néhány vektörny
- jái vár le man.
- be (1/0 eqpége) " : lyulskáltya
nyantató
- 256 lytos alap töltőprogram
vált kóttalárisen ar hirta le
a lióramfészélel pirogilete
- fáváskádát kelyskárisóttaláris
válláqva vagy kóttáris

- fordítás: eredményt szintén leírja
 tatta: hibajavítás: hirtelen megáll
 egyre csak egy rakasát várólak
 le, azt fordítja

egyre több hibával jött: a tárgy
 hibát módosított, abban ugyan

- CPU kihasználás: 7%
- átlagos sebesség: 4 job/h

closed shop rendszer: a programozókat nem engedték
 be a gépterembe! csak kikapcsolható operátort
 nyúltak a géphez. Hibával dump-ot
 nyújtottak a programozóknak a memóriáról,
 elmentve a hibát.

job - job leírás nyelv

melletti vagy
 kérték
 kérték

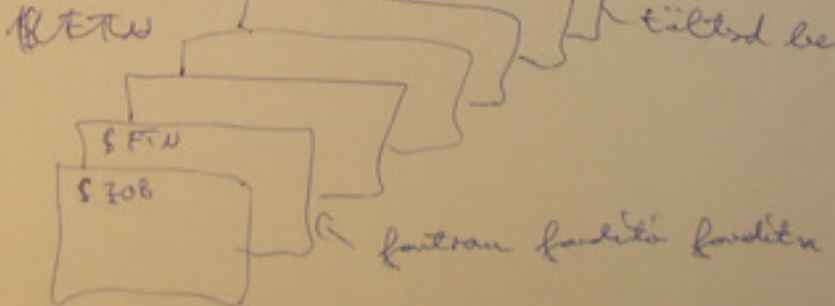
megvárja a programozó az operátor-
 nat mit kell csinálni, egy formális
 nyelvben.

tegyük le az operátort a
 "gépre"

"Egyre" monitoros rendszer

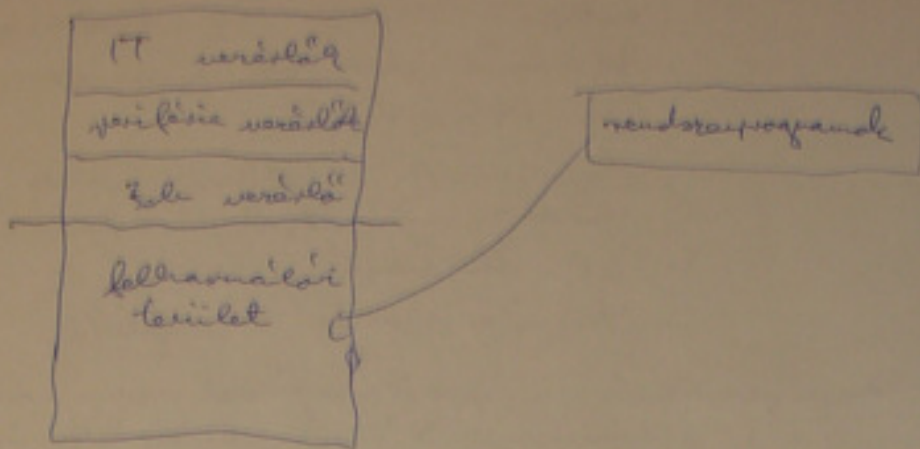
"Egyre" monitoros rendszer

- adatfolyam
 vereslőjellel → kód → feldolgozandó adatok
- §-jellel kezdődik a vereslőjellel, vereslőjellel
 véli: §END



1/4 ↓ azt kívánják megismerkedni a géppel,
 attól kezdve nem kell operátor +

Menäsiälyp:

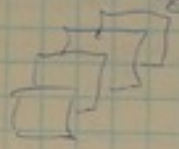
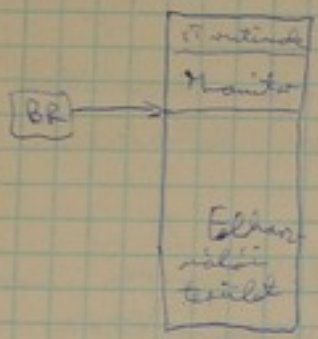


- CPU kiharumäläi: 55%
- Itäimäi kaperiäi: 33 jöly/a

overhead: rendszeradminisztrációval töltött idő

benchmark: teljesítményvizsgálat program

egyszerű monitor ← operációs rendszerek



gátlószűrővel szembe

"Ha hibás a job, megkérheti a következő job elvégzését adatként."

1/3 - védelem kérdése

Védelemhez hardware-támogatás kell

Monitor: nem minden szabad

Felhasználói job: korlátozott

Legalább 2 speciális mód kell

(1) Rendkívül / Privilegizált mód

(2) Felhasználói mód

- privilegizált utantárolás
- illegális utantárolásokról való ellenőrzés

Rendkívül privilégizált utantárolás

RM - hátrányos beállítások a felhasználói felületen keresztül

- megvalósításukról a monitor végére

- csak az agvendoren device driveri elvarhatja a kistozalant => nem elvarhatja le felhasznaloi program mas zoli kistozant
- 1/0 utantarsol is legyen privilegizaltal
- megvaltasinal vendor-moddia kavesal-gon a proc
- IT - utinali vezetatalija monitor tomlaten kell legyen, ne lehetson ativi
- megvaltasini vendorat le proc modot alliti utantarsol privilegizaltal
pl.: felhasznaloi zoli ne maradjon ki az IT-ket.

...
 MASK -> innen csak
 HLT RESET - tel
 ...
 ࣘ
 pl.: HLT is privilegizalt

CPU - kiharualas = 55%
 atenerstohizeseg = 33 zolo/le

Hol a maradék CPU teljesitmeny??

Lassa periferiak!!

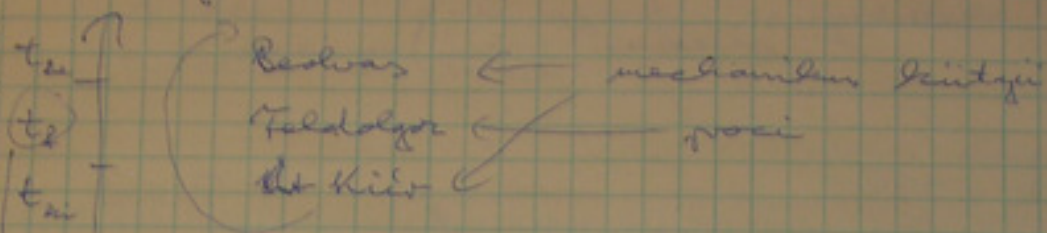
Selvenyintal:

Elektronikus eszkialak: kaggarable
 (ms...ms)

Mechanikus eszkialak: (ms)

Gmbel / kiharualo : pl.: kullenyfizet
 (ms) nemetpanel
 kaggarable

Magyarozások:



három CPU-ik

1 input utantár ideje alatt
1000 más utantárt tudna
megkezelni a proci

I/O idő alatt holtidő!

$$CPU_2 = \frac{t_p}{t_m + t_p + t_n}$$

Sálkaszámítások a CPU működés a mechanikai
szériák működésével !!!

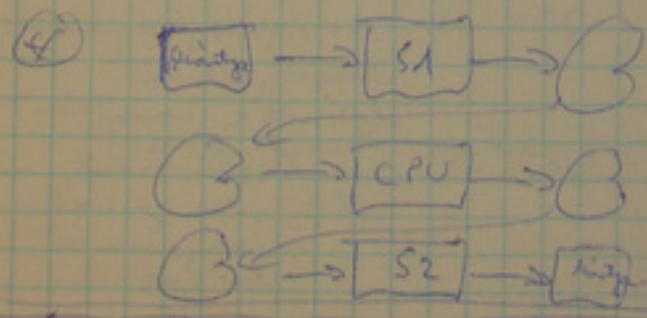
lényeg megvalósítás fogja jellemzők ha jók az
adatok átvitelének

benne van működés alatt fűrészes márkéjé!

Offline I/O műveletek

széles - feladatokról - más párhuzamosítás

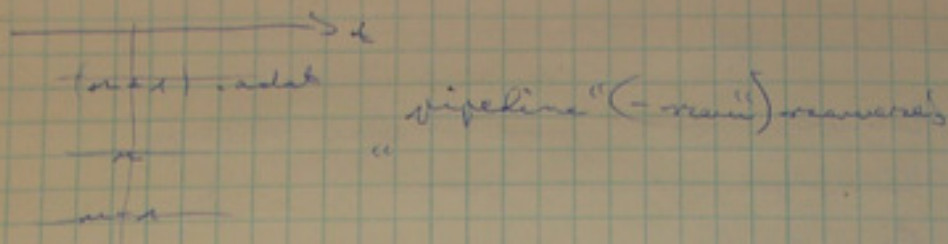
Megvalósításuk ha main felgyorsult, a CPU-val
összehasonlítva a sebesség elvárható az adat
kezelés szempontjából.



CPU-val
szériák szériák
dolgozva

S1 } márkéjé
S2 }

magyar "hangszámszámítás" 1 fordulat 1 sor



$CPU_{\text{eff}} = 30\%$

$A' = 55 \text{ jobb/er}$

gond: a program úgy van megírva, és
 számításvárakból áll és is megantatva
 megantat

DE

itt valahogyan alvadás és valahogyan ivak!!!

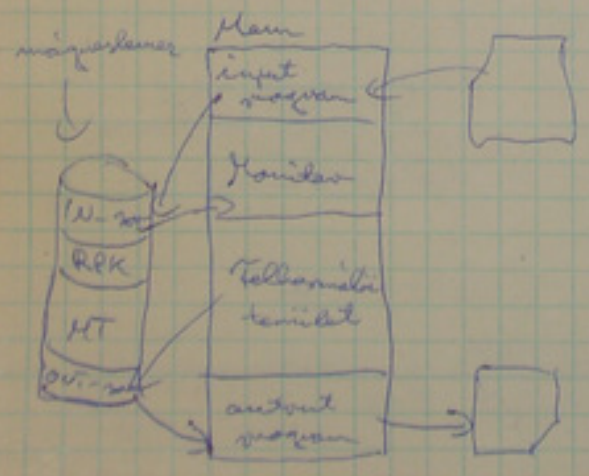
|| igény:

kiszámításfüggően /0 programozás;
 logikai perifériák bevezetése, ami megajta
 teljeskörű perifériai kiszámítás lehet.

* Ezt a tervezést oldjuk meg egy géplel!!!

SPOOLING

Simultaneous Peripheral Operation Online



RPK : rendszerprogramok
 lementése
 pl: itta compiler

MT : memóriaterület
 pl: ide kerül a
 compiler a
 kódok

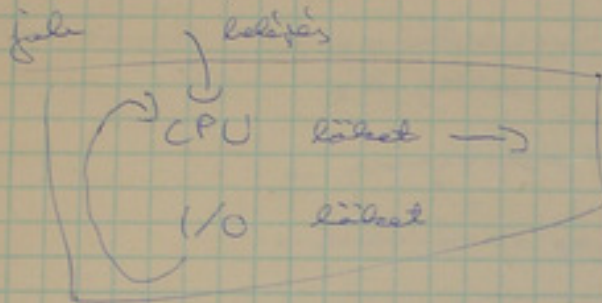
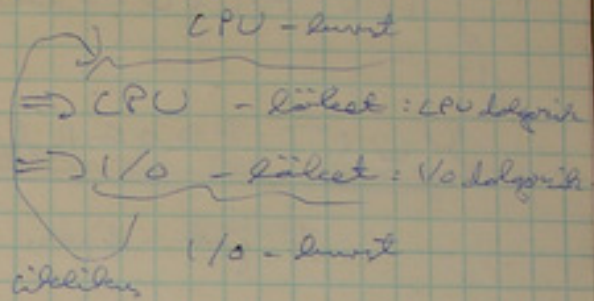
W/O - queue - val
 gyantitjuk a lassú
 eszközöket

A kórné I/O-ka csak a megvalósítás idején
tevékeny a pocsol.

GAAP: - megismerés - mem - CPU - I/O
szórti szabályozás

- Interaktív felhívás?

jól: CPU - halmaz
I/O - halmaz



Programok: - vannak amik sok I/O és
kevés CPU - időt / műveletet
használnak pl.: konvertálás

- fordítva: pl.: egyenként megoldás

I/O-intenzív } programok
CPU-intenzív }

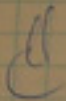
Kiszármaztatás: I/O & CPU művelet
időigényi kiszármaztatás

||
optimális jól-mű :-)

P_i : Szervező nem vektorizálható, I/O -queue
 és O/I -queue is $R(I/O)$

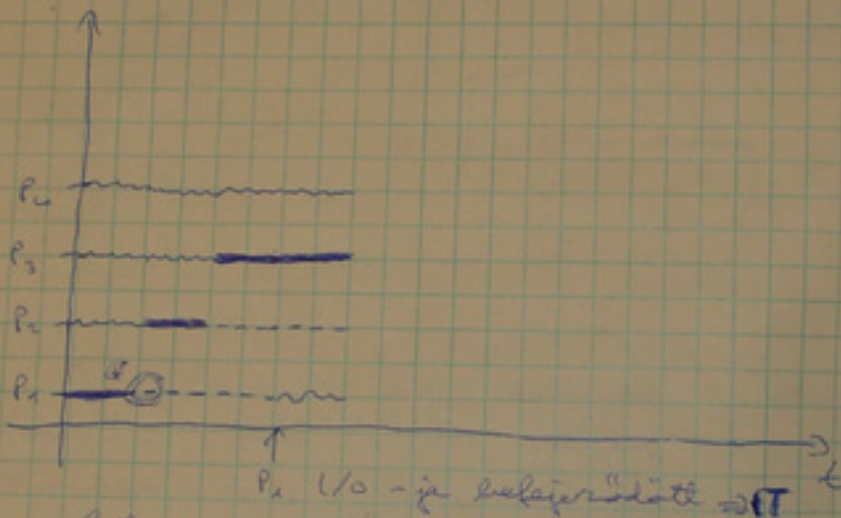


Körvetlen a magyarságra való vezetés a
 jelszót egy alprogrammal, ami betöltés
 I/O - CPU - idős jelszót választ.



MULTIPROGRAMMOK

\forall jelszót i programminőségét P_i



— : fut

— : tart futni, ha van CPU-n

⊕ I/O művelet kész \Rightarrow várja az I/O művelet
 befejezését; fenn van még az I/O listán
 alatt; így az idő meg lehet tartóztatni
 a folyamat más futásáig

--- : alsó, vár



döntési pont: P_3 -at folytatni,
 vagy inaktiv?

pre-symptome-intervall: P_2 -t kezdésben
 P_1 -re ha kezd az I/O.

P_2 -t nem azért szabadon meg-
mát a sebesség \Rightarrow bizonyítás

Ha sok program van: a CPU mindig kap
munkát.

↓
válogatás a mik-ben \Rightarrow teljesítmény-átvitel
mérés

- Problémák:
- analízisben az elterjedt programok
állapota
 - programok közötti gyors átvitel
 - memória-gazdálkodás
 - I/O kezelés: I/O műveletek sebesség-
mérés
 - koordinációs problémák, holtpont

(DICK) Előadás

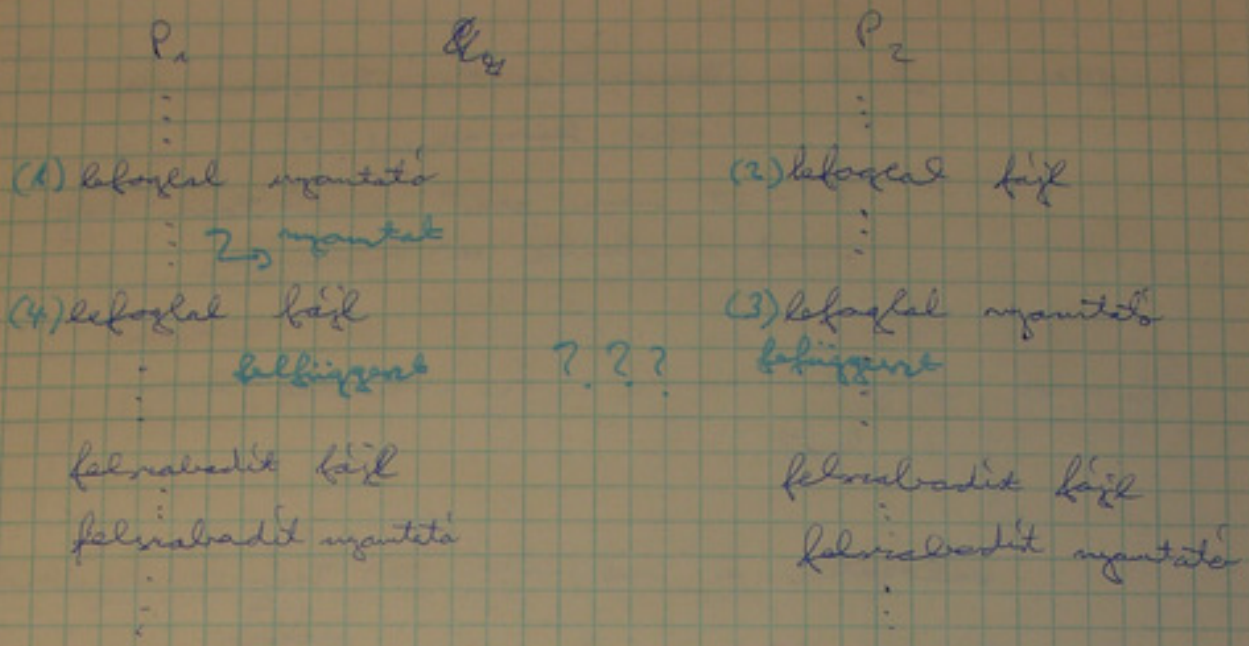
11. ok.

- Itt kell eldönteni, hogy a folyamatok ne
indulhassanak el egyszerre n. az az eredmény.

↓
válaszadási lista

- folyamatok közötti átvitel megoldása
- egy erőforrás egyszerre csak egy folyamat
használhatja \Rightarrow rendezési lista;
spec. műveletek

→



holtpont

P_1, P_2 futása elakad.

- o -

Itt nem csak az operációs területet kell védeni a programtól, hanem az egész programoktól is egymástól.

Jobbi védelem kell.

MULTICS

1/60-as évek, Boston, MIT;
 laboratóriumban kóroval → 1 db nagy gép
 erőforrásigények ⇒ lemondás #/

UNIX alapjai (mellektémle :-)

Multiprogrammierung die Grundlagen (wie es ist)

1. Ebene: globale Anweisung

Batch - umgebung

→ Algorithmen = Job
Jobkette

2. Ebene: idiosynkrasie, Synchronisation, Terminale, Hardware

1. Job, Teilbereich ⇒ Multiprogrammierung

interaktive, idiosynkrasie, Synchronisation

• Belastung
• aktive
• idiosynkrasie
• idiosynkrasie
• idiosynkrasie
• idiosynkrasie

• idiosynkrasie
• idiosynkrasie
• idiosynkrasie
• idiosynkrasie
• idiosynkrasie

(ON): idiosynkrasie, idiosynkrasie
idiosynkrasie idiosynkrasie
3 - 6 idiosynkrasie

idiosynkrasie idiosynkrasie idiosynkrasie idiosynkrasie

3. Ebene: Real-time (aktive) Synchronisation

Job = idiosynkrasie

idiosynkrasie idiosynkrasie idiosynkrasie
idiosynkrasie idiosynkrasie idiosynkrasie

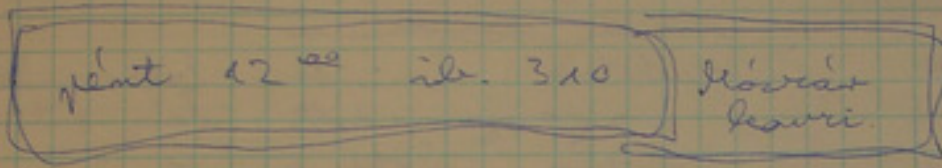
idiosynkrasie: idiosynkrasie idiosynkrasie
idiosynkrasie idiosynkrasie idiosynkrasie

idiosynkrasie idiosynkrasie idiosynkrasie
idiosynkrasie idiosynkrasie idiosynkrasie
idiosynkrasie

idiosynkrasie idiosynkrasie: idiosynkrasie

Grundidee mindig 1 programmal (ill. jövedelmével)
bármikor, ezeket használjuk a feladatunkon
áttekintésre.

folyamat = egy program végrehajtása;
végrehajtás alatt álló program

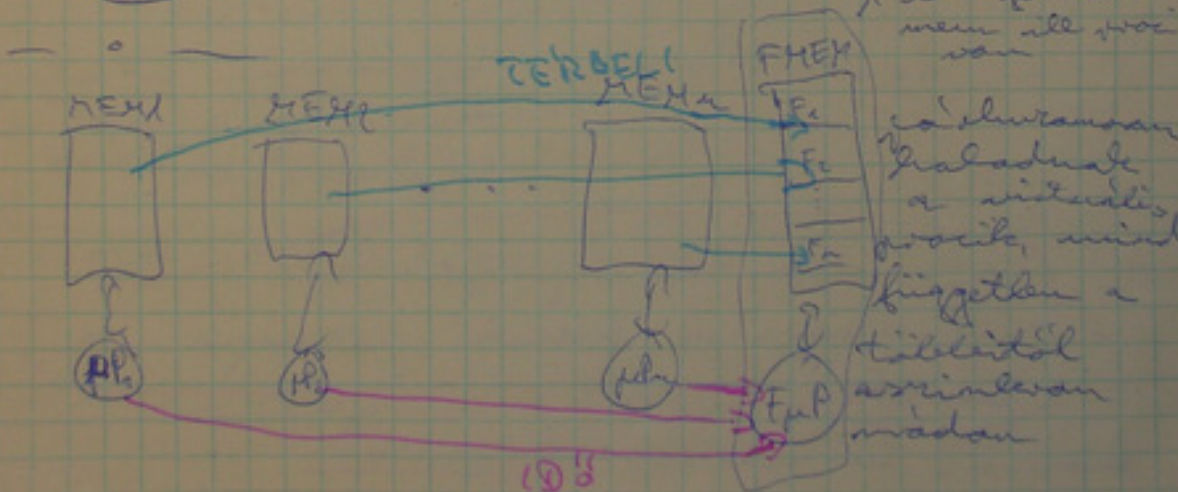


~~szó~~
ny programoknál elvileg lehetnek!!!

folyamat (modell)



proci állapot: hol tart a folyamat
▶ állapot visszaállítása:
→ azonos utasítás már lekezdte.
Több, az új még nem
bevégeztette.



(10) 3

DE lehet kapcsolat:

1 versengés
pl.: erőfeszítés

2 együttműködés

pl.: közös feladatot hajtanak végre
információval

Opvondres:

virtuális értékelést hozzárendeli a
fizikaiakhoz (MEM, MP)

MEM: területi lefedés \rightarrow RHE-modell

MP: időtartamos lefedés \rightarrow szoftvermodell

keverék: ha nem fér be a program a tábla:
teljes & időbeli lefedés
keverék

velemények osztása
read: write
intézkedés a vélemények

intézkedések,
vegyesek

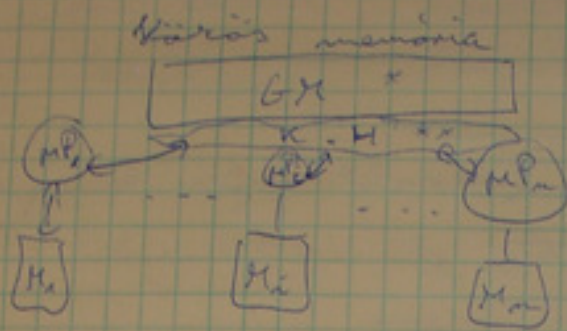
nem privilegizált
kezelés

privilegizált
(folyamatok RHE kettős
nézőpont az opvondres)
privilegizált
csak az opvondres
használatja
rendszerhíváson
kezelés elvégzés
el

Opvondres jelölési a, vagy intézkedések.

folyamat: nem privilegizált vagy intézkedés
+ opvondres rendszerhívásai

Esquithumilitades



* ofobaliv memoria
 ** kaperababalarat (min)

GM elise:

read (cim, adat)

write (cim, adat)

lemaia
 paramita

lokaliv mem.: { read
 write
 mem nitkardet

glab. mem.:
 kicavrite:
 mag lokal mandani
 mi leqen nitkardet
 minelatek ereten

1 read-read
 adat nem tovul,
 minden always
 u. art leavja

2 write-read
 ↓
 kelihi

irenet-tavalitisa
 esquithumilitades



xxx adabalarat

1/0 erikendit latjak
 esqvast irenetovakilitas-
 nal

send (...)

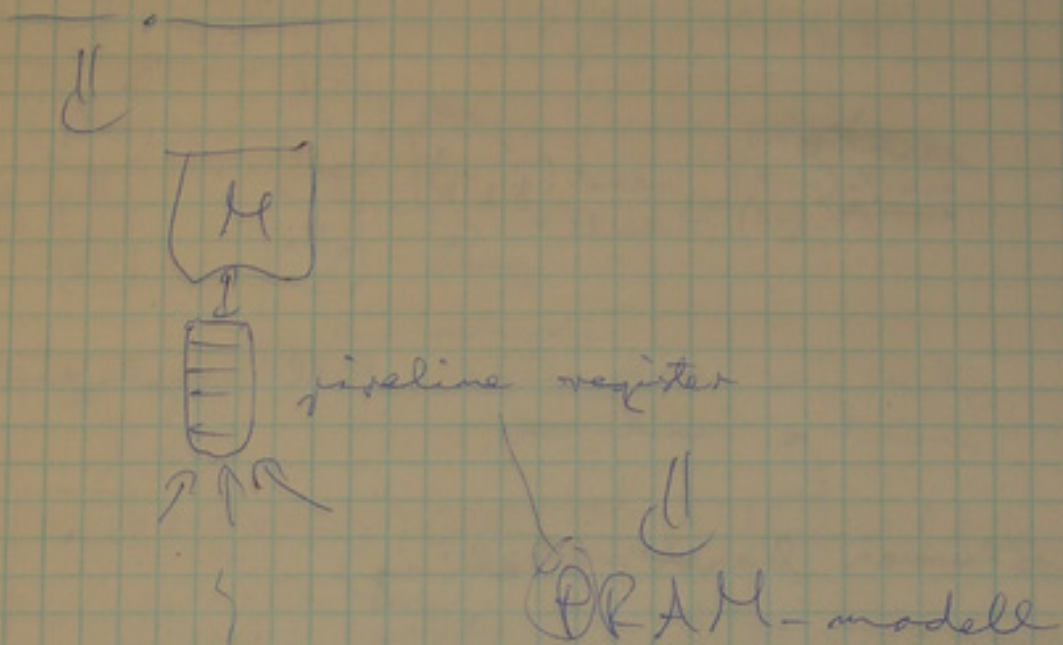
receive (...)

na leqen istaferencia

leokentlo kelihi titat VF
 VGB ar clavo VGB ar ni sekeritakmat
 olvava

3. suite - suite

elvárás: egyidejű végrehajtás legyen
& LRU az egyik lehetséges feltétel

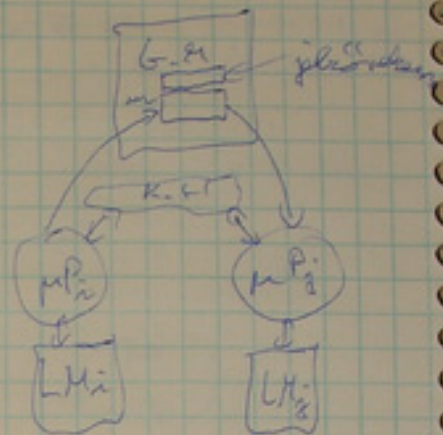
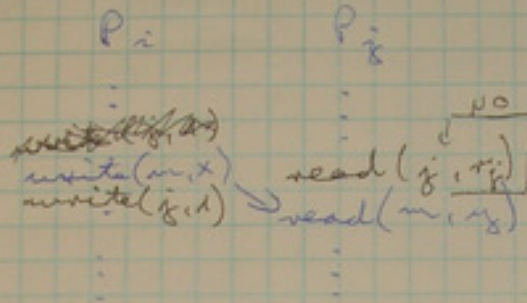


1. mem. vétele dönt a sorrendről (végen)
2. multi-masteres szinkronizáció, ahogyan a szinkronizálás, az is a FIFO-
(ma) -ben

Ha legyen ábrarajlás:
"jelöltve" használata

Szinkronizáció

- precedenza (sorrendiség)



A write - más olvasás kell megelőznie, mint a read - más.

software hand - shaking

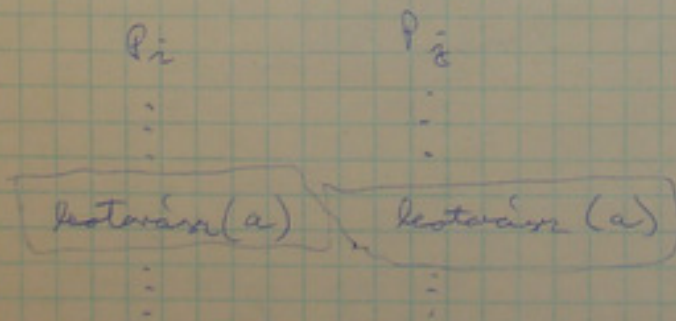
előrel lépésenként kizárólag az egyik gépnél

titkán csak végsőként tudok, de a jelölés csak egyike esemény.

- kölcsönös kizárás

ez is a PSM modell probléma, csak több vektorra.

vektor $a = \{ \dots \}$



← csak folyamatosan az NEM használatuk vége

- egyidejűség (rendezés)

útszám alapján egyidejűségű kódokkal számolt.

A feladat megírásak egyaránt

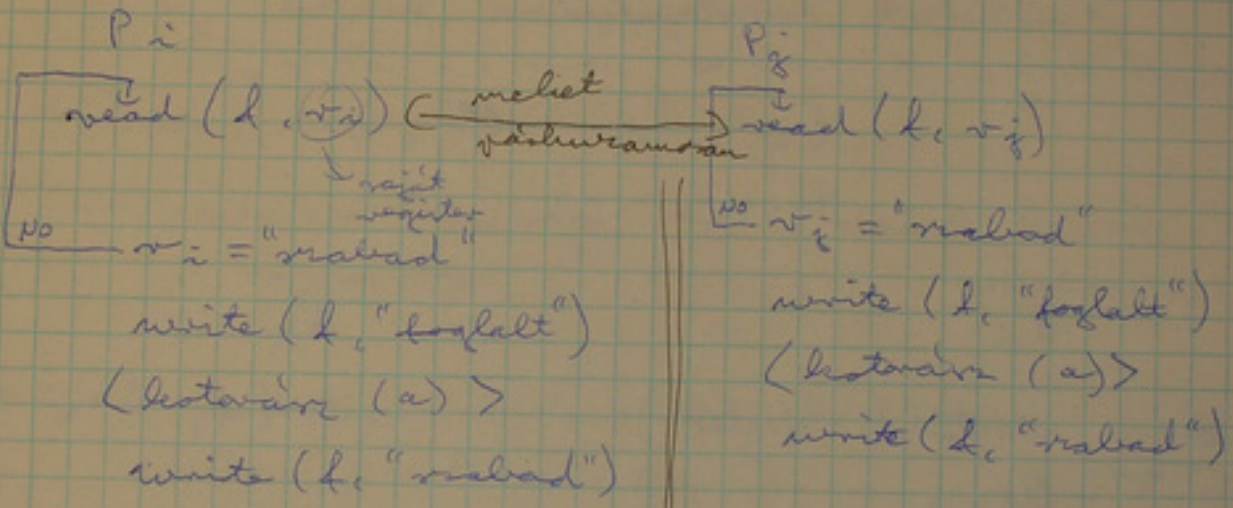
Közlésről leírás megvalósítása

①

G.H.

l : flag

a : szöveg



Mindkettő szabadnak látja
 Mindkettő kezdésével kezdene.
 Ritkán előfordul hibák DE veszélyes.

* Itt megvalósítjuk a megvalósítást, de más
 ne kezdésével. De ez modell
 szinten nem megoldás.*/

↙

↘

Dijkstra

↳ 60-es évek
 algoritmus, ami n^2 folyamatosan
 megoldja a problémát. Ez felt.
 másként $n \log n$ -es idővel
 megoldható nem használják

Dijkstra - féle algoritmus
 első használják

Petersen - algoritmus (1981)
 első használják

$f_i, f_j = \text{klasz};$
 $\text{tumor} \{i, j\};$

P_i

kezdés: write(f_i , "foglalt")

utána
 write(tumor , "i")

megnérem
 más
 festőművele
 read(f_j, v_i)
 $v_i = \text{"szabad"}; i$

megnérem
 más is
 kész - e
 engedélyt;
 nemem van
 read(tumor, v_i)
 $v_i = i; i$

itt kezdés
 lesz

P_j

write(f_j , "foglalt")

write(tumor , "i")
 read(f_i, v_j)

$v_j = \text{"szabad"}; j$
 read(tumor, v_j)
 $v_j = j; j$

2. lépés: közös változó,
 az ugyan, ami elölük állt

kezdés: write(f_i , "szabad") write(f_j , "szabad")

u feladat végsőre

Baksey - algoritmus (minden P_i feladat
itt való is kell ^{invariáns} a ^{egyenlet} feltétel.

gond: kooperáció kellene, de csak
vadász és nyúl között van.

- a megfigyelő megállapít magának egy
sorozatot, ezt követően egy táblázat a
memóriában
- új életről nem vizsgálva a táblázat
megfelelő a más sorozatot, ezzel egyet
megjelölve az magának
- megfigyelő az első állapot, ahonnan kiindul
a sorozat. Ezt követően, vizsgálva a
táblázat
- feltételek mindenki legyen a sorozat,
aholha járható in, ha nincs valami kiindul

GOND :- más megfigyelő a más, de egy
valamennyi járható állapot hamarabb
eléri az 1-essel megjelölve

||

1 sorozattal 2 kör

- végsőre van, végsőre, de milyen
gyorsan vérsi fel k területe a
sorozatot

||

Ha :- kell egy jelölés, de valóban egy
sorozatot választ. Ezt ki kell
választani.

co

- Ettől még 2 egy forma szerint lehet
No. - a saját aránytól lent és a
saját egyenlő.

formán és új alapján is hasonlítok
arányt

...
Am a világ más utat választott.

...
Guns are coming soon.

Kalman I.B. 310

U max

Ka 51

K1 20

K2 41

} lemondásról a dalon
intérend !!!

— o —
Kettősnyelvűtől valószínűleg a folyamatok egy
néhány megvalósításának megvalósítása.

↓
CPU intézkedés.

— o —
Kettősnyelvűtől valószínűleg a
multinagyvonalú rendszer tudott
valószínűleg.

RISC: allt namn multiplicerats av, namn
är bärare

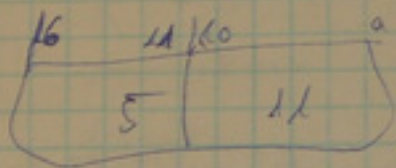
16 bitar skiljda av

1 M byte - va adresser härbiter

32 var är index av tillbehör

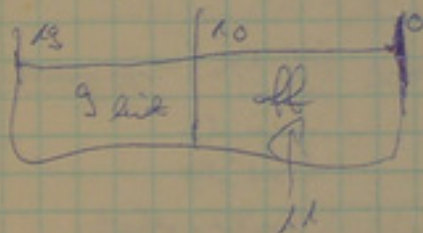
1 adress bit

flera bit är skiljda?



offset: 11 bit

flera bitar, a register 1 register?



tänker 1 register (9+1) = 10 bit

3 2 register all a block

Bekeny alq. => hanyalult!

terve a jelcöbölthet??

Kell egy övretett elemi utantás, ami
hiszlen és vissrait egyben.

I. Mo.:

Text and Set utantás

Hiszlona a flag eredeti értéket és
vissrait egy márkat egy műveletben.

Definiós program:

TextAndSet (F) // F flag e parameter

begin

TextAndSet := F;

F := FFl; // haan FF-et tölt be; wrap

end

örllhatatlan
művelet

Egy új utantás =>

PR19C - modell:

- read
- write
- TextAndSet

memória-utadás:

read (F, r)

write (F, "FFl")

örllhatatlan

Nequalitás joci valad függő.

/x read
write

egymás után márkalt (i - bel u. sz)

Multiprocessoros rendszerben

pl.: VME: read-modify-write ciklus

↓
átvétel

Intel: LOCK, UNLOCK utasítás
ez egy blokk, a lock után
kezd utasítások alatt nem engedti
át másnak a részt

II. No.:

XCHG - orsz

2 memóriahely cseréje szinkronizálással

XCHG(LF, GF) } flag
var TEMP }
 ↑ globális

```
begin
  TEMP := GF;
  TEMP := LF;
  GF := LF;
  LF := TEMP;
end
```

} szinkronizálás

lock:

~~lock~~

P₁

F

P₂

szint
szinkronizálás
folyamatosan

r = terület(F)

r = 0

i ↓

(memória) // memóriahely

write(F, "0") // valószínű
irás

sz. így.

Task 2:

GF

LF1

LF2

~~write(LF1, "FFa")~~

~~write(LF2, "FFa")~~

XCHG(LF1, GF)

XCHG(LF2, GF)

LF1 = \emptyset

LF2 = \emptyset

i ↓

↓

helyes

< mutat >

write(GF, "a")

// szabadra állítás

// vámside

Helyes levezetés voutjaka
női teljesítmény

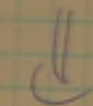
Multiprogrammál helyes jár
a női, levez, és nem adja oda
a CPU-t márnak, ha nem vessem
el a vámsidekkel.

Multiprogrammál n. egy nutesidek,

amig levez, altarnak el! Majd
felévesitjék, ha tényleg van.

De egy operációs megvalósítás,

Maig yord: Nincs igazság, és mindenki szövegesül. Lehet olyan, aki végig tanul és nem tanul, azaz hacsak feladattal.



Fair intézkedés a no.

Mindenki szövegesül, addig ismétlődik nem valójában, hi van az élet, míg az élet kivétel nélkül.

3 - félév szövegesítés igazság precedencia
 felismerés kezdés
 kezdés, egyidejűleg

misszoriális
 szövegesítés
 élet: SZEMLEFOR

Gramma (Diktóra)

"vanti jelszó"

hét este: kezdés V. tilos

Definiálás program

var s: integer := 1

1. while P: begin

while s < 1 do write;

/* amíg s < 1 ilyen jelölés

* kezdés

*/

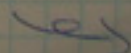
s := s - 1;

end

szövegesítés

2. while

V: s := s + 1; // szövegesítés



2. Alapfogalmak

P : tud helyben járni

V : érintetlen, jelszóra a táblázat, h
már jelszóra

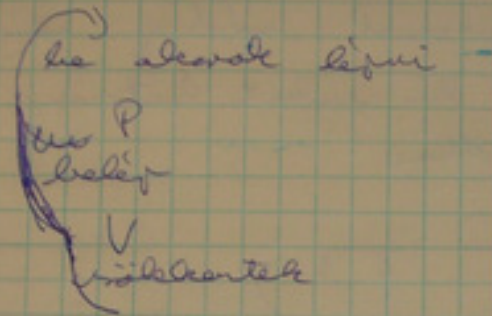
ha 5 helyszínt akarok beengedni,

$k = 5$ a kezdőérték

helyi
móddal

\forall helyszínel $P-t$

\forall helyszínel $V-t$
ha nem megy



amíg más nem
szól be újat

ha volt az általános szabály

Bináris nyelvek

$\Sigma = \{0, 1\}$

- x 1 valódi és tud menni,
- x utána mindenki
- x felmenni, míg az az
- x egy ősi nem megy
- x/

Precedencia megvalósítása

$\Sigma = \emptyset$

← minden nyelvre

P_1

P_2

\vdots

\vdots

\vdots

\vdots

\vdots

\vdots

\vdots

\vdots

U_1

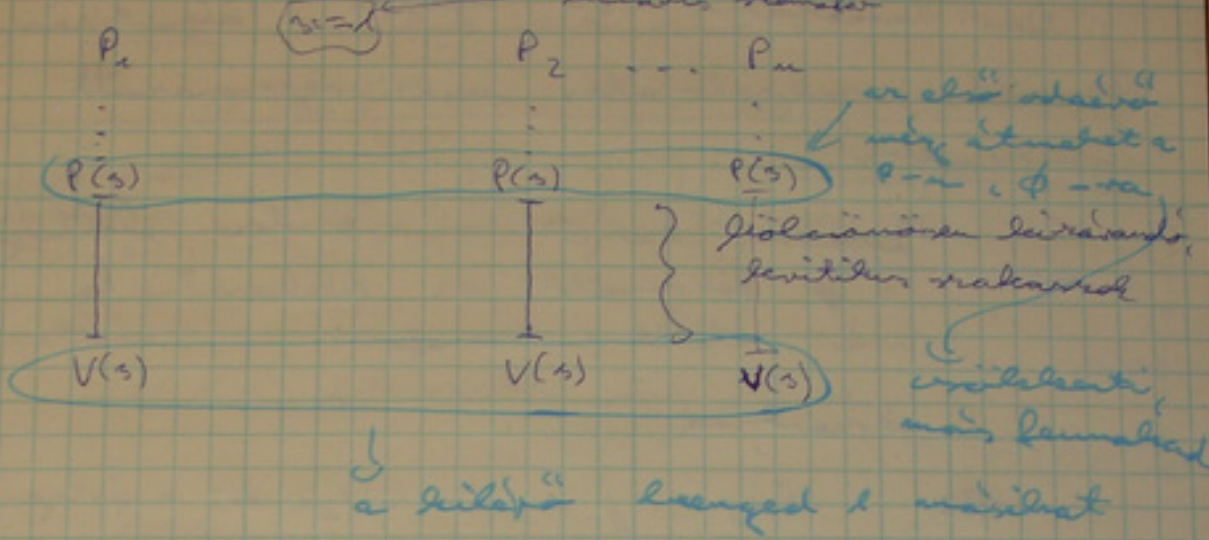
$DP(\Sigma)$
 U_2

vár amíg a
nyelvre változ
évtől U_i feladat
szóval
nem lesz \emptyset -mal

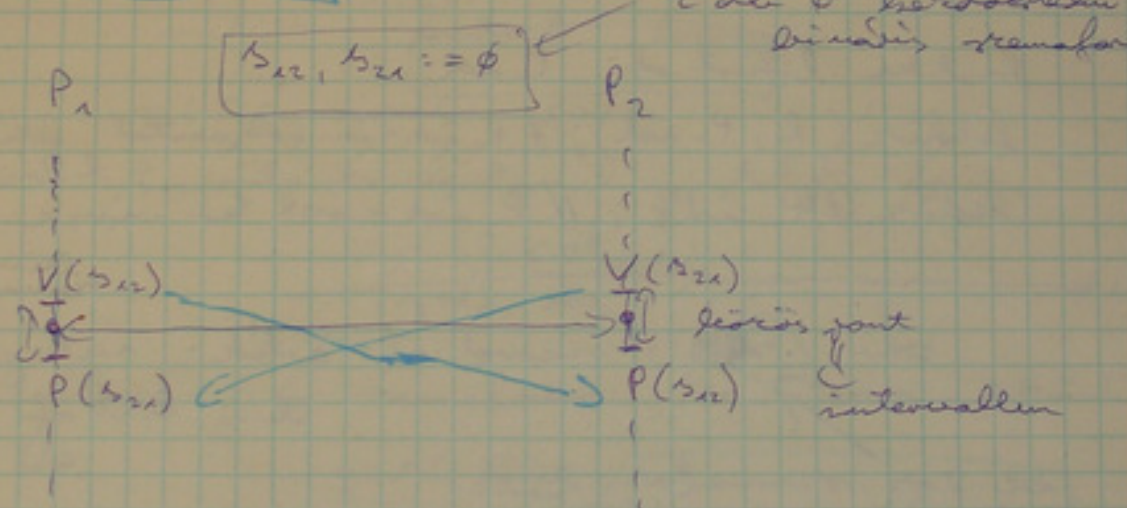
$V(\Sigma)$

→ táblázat

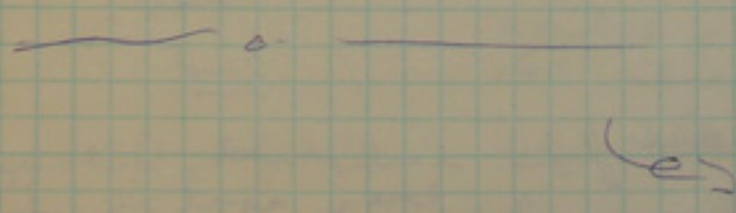
Kölcsony leírás megvalósítása



Randevő



Jutnak neki megjelölés az intervallumban,
 való a másképp, h itt van, leírásuk
 pedig, h leírásuk.



Heur listat megvalasztani a memóriát?
 Van listák a CPU helyen? Felismerés!
 Memória megvalasztás multiplatformon
rendszeren

Dijkstra helyett az operációs rendszerek
 a \neq folyamatok és felismerés
 más állít a memóriát

2 rendszerrel

~~sleep~~ sleep; // maxime adja ki a folyamat
 wakenup(Pid); // operációs adja
 ki a folyamat

lista: list of Pid

felhív (l, P-id); // P-id-jü folyamat felhívja
 l listát

felhív (P-id); // P-id-vel ter vissza

listás
 memóriát

var ~~...~~ = record of state: integer; := 1
 l: list of Pid

az egy memóriát end
 definíció, megvalasztás
 procedure P(~~...~~): // memóriát kiadható
 P művelet

OSZTHATÓ LLL
begin s. state := s. state - 1;
 ha a memóriát if s. state \neq 0 then begin
 váltása < 0 , felhív (s.l, (s. state))
 elalvók, DE
 jellek maguk, sleep;
 a elvárások fel, end
 felhívom magam
 P id-vel a listán
~~...~~ // ha kiadható, felhívom
 a then azaz is vég

procedure V (~~2016~~):

lesqin n . érték := n . érték + 1;

if $n < 1$ then wakeup(lefir(n, l));

∇ váratosa
csillásként az
értéket, az
átmenet end
negatívba.

// egyelőre átugrom és vár

Ha megyl 1 => felélesztés elint

OSZTHATILAN

Fair-e ez?

Attól függ, ha a listakeresés
melyen listát kerel.

FIFO - nál fair

Priorityos listánál NEH fair.

($n+1$). ^{Első} ~~le~~

11.11.

Erőforrás ← valamelyes használati példák

↳

request(iD)

release(iD)

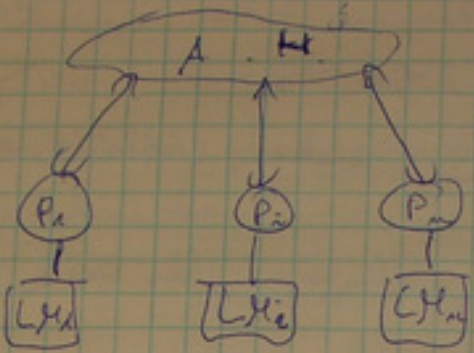
Erőforrás ← váratosa, váratosnak a beírás/leírás,
beírás/leírás ∇ váratosát
elindít (szemaforál ellenőrzés ami
való 1-et enged be)

Törlés az elöllet, iméntől
keresés váratosa a beírás/leírás,
aktiválás.

delay - váratos, idősít
felhívás

váratos: egy generál erőforrás várak

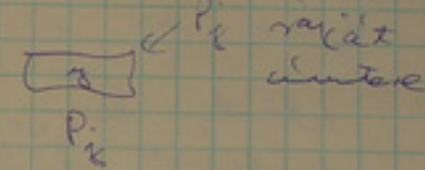
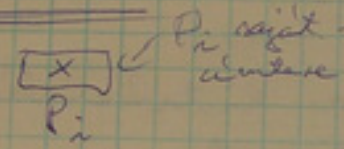
üzemeltetési módok, számítástechnikai adat hálók



send()
receive()

- megnevezés: hosszú valójában nagy számú a folyamatok
- szinkronizációs hálózat
- szemantika: mindegyiket szemantika

Hogyan néz ki



$y_i = x$ nem működik

Díjazt megnevezés

send(P_i, x)

receive(P_i, y)

- // Hogyan néz ki a hálókészítés és a hálókészítés
- // nagy számú, ha a hálókészítés $P_i - x$ jön
- // melyik más is, a hálókészítés

Indikátor

send(HB, x)
(CH)

receive(HB, y)
(CH)

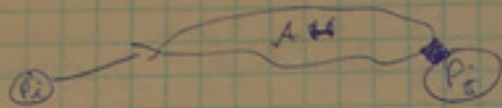
csatlakozás

- // Hálókészítés nem díjazt a hálókészítés, csak a hálókészítés
- // nagy számú a hálókészítés

szinkronizációs

- // szinkronizációs, a hálókészítés a hálókészítés
- // van a hálókészítés

Input part

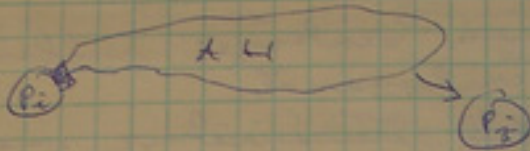


send(P_j, x)

receive(y)

// "oraz egg
// kimeretel lat

Output part



send(x)

receive(P_i, y)

// "oraz egg
// kimeretel lat

pl.: Klien - server modell

pl.: Master - worker modell

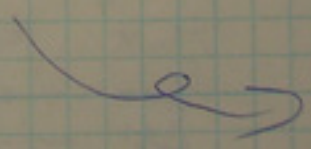
magyar
a munkat
vagy szolgálat

algoritmus, ismeretek, feladat a munkat

Magyarítás

Broadcast = küldni mindenkihez üzenet
küld

{ 1-mas terven vagy a címre az adott
üzenet mindenkihez → üzenet
váltakor teljesítés



Geometriai algoritmusok

egyetlen lépés, minden lépés után

minden lépés a legkisebb \Rightarrow precedencia

minden lépés a jelen \Rightarrow rendezés

// szűkített lista után rendezés

// kiegészítés

egyetlen lépés \Rightarrow nem szükséges szűkített
listával

1 db menüvel vagy
sorokkal vagy listával

lemezestét a "lemez" is, nem
tehát iránt a szűkítés

elválasztás: FIFO szűkítés

regisztráció szűkítés \Rightarrow az első lépés a
lemezestét, mindig lehet
az iránt a szűkítés

↓

lemezestét \Rightarrow adatvesztés \Rightarrow

\Rightarrow hibajavítás

Geometria

Van megfigyelés az ellipszoid iránt?
Geometriai - e az egyenlet?

Formai leírás? Újra szerkesztés,
tárcsák? \Rightarrow iránt ismétlés

Helyen alattam lévő iránt leírás?

Hitt tud az iránt átírtak vége?

Boardszámla vizsgálata?

Attól minél? Attól az iránt leírás!

(e)

broadsheet, nem pontosan képződik és
váltakozóan képződik.



Égypárvé képződik és várak mindenképp
visszajelzésre. A legfontosabb visszajelzés
száma.

Sud - Recive : folyamatátvitel egyik oldal a
másikra

csak
gyermekként működik egyik
munkatársak a másikkal =>
=> csak a munkatársak

szóval:

Ha sud után megváltoztatam az
ellenőrzött változó értéket, az már
ne változtatna az ellenőrzött változó
értékét.

Többi folyamatok általánosítása

1. A folyamatok külön programként
írunk meg és az a programok
futtatja multi-programozással
Nincs nyelvi támogatás csak
ajándékosított.

2. A nyelvnek leírásai vannak amik
leírják a programozásról a fűtő
árakat

pl.: Java

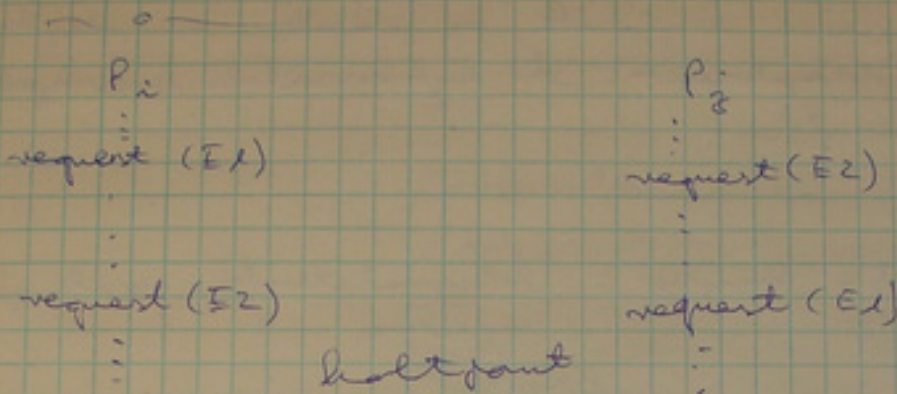
Leírás - Pascal

Java

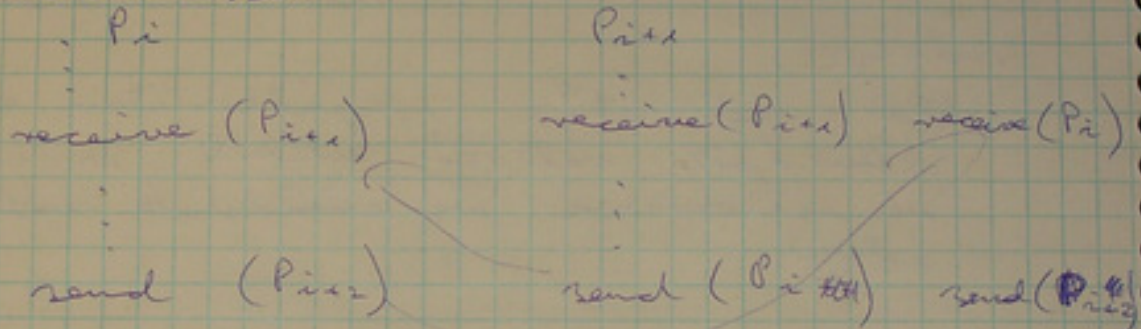
Leírás
nyelvi

=> ehhez programok a megoldások

Haltponthelyzet



inverzió sorrend:



ez is haltpont helyzet

Haltponthelyzet: A rendszeret állható folya-
matok egy H-ből vártá akkora van
haltpontban, ha \forall H-beli elem
olyan eseményre vár, melyet csak
egy másik H-beli tudna
elindítani.

akár: előfordulhat
nem fordulhat

\forall várakozási helyzet

Központi helyre kiadások:

- request : - ha későbbre is tudjuk halasztani
a követ. megteremt. befejez.
- release : - ha nem fejezték a folyamat
folyamatát követően

Erőforrás megnevezés

erőforrás \rightarrow menthető állapotú
vse - szintű határolt pl.: CPU
memória

\rightarrow nem menthető állapotú
nem vse - szintű határolt
pl.: nyomtatás

erőforrás \rightarrow egy példányos
request (E1)
release (E1)

\rightarrow több példányú védőanyag
pl.: memória

request (R1: X, R2: Y, ...)

R1 -ből X-et követel

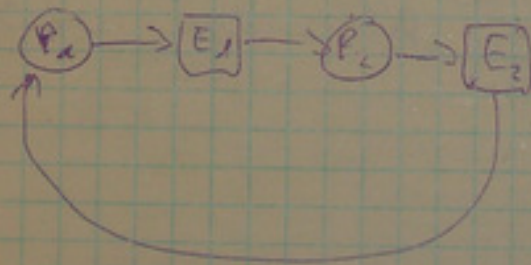
~~request~~
release (- " - ...)

Ha a folyamat csak akkor foglalt,
ha a H erőforrás egy példányra
megfelel, különben váratlanul kell.



Feltétel:

- legyen kölcsönös irányú a kölcsönös
kapcs. nyelv erőforrásaitra
// legyen nem menthető állapotú erőforrás
- állapot & utait legyen
↓
szimuláció más (vagy egy) mártva (erőforrás)
(erőforrás)
↓
↑ ilyen helyen
- Ha preemption
erőforrásra rendelték nem veszik el
erőforrást
- töltés utait
leőlevezés
grafon szemléltetett erőforrás - foglalt
- van DE legyen két lépés
el: foglalt & várakozás
visszatér: erőforrás



Haltfont keresési stratégiák

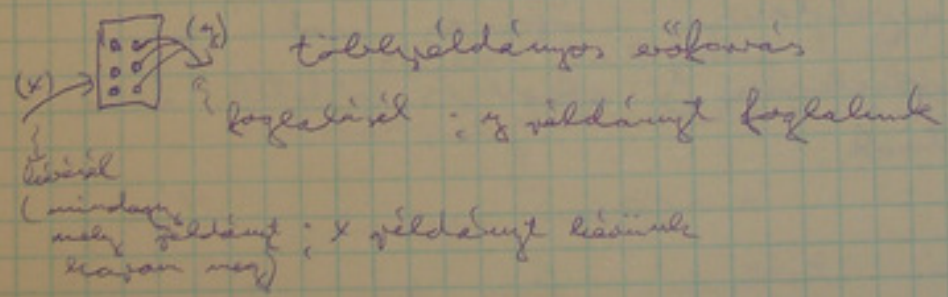
- ① Greedy algoritmus
nem tökéletes a problémával nem lényeg
nem is meg a szükséges lehetni
- ② Haltfont megelérés
↳ eleve nem is lehet haltfonta
nemdeterminált; területalaki
haltfontmentesíté
- ③ Haltfont ellenlérés
- ④ — " — (értelemszerű és feladatszerű)
↳ dinamikus, futási idejű
ingyes előfordulás - kivétel

(4.2) Góddás

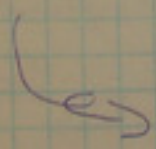
11. 13.

Előfordulás - foglalatlan gyakorlat

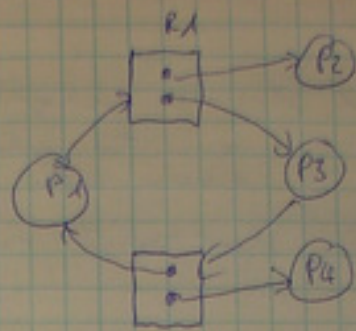
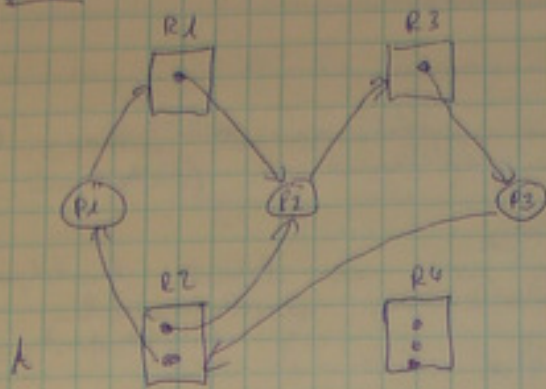
- feladat
- előfordulás
- → □ le van az előfordulás, de még nem lehet meg
készen
- → ○ foglalatlan



(minden
meg példányt; x példányt készítek
házon meg)



pl.:



Pi megalayja ar evofonast: kivésit \Rightarrow loqalásél
li kalvadaditja — " —: loqalásél kitévéládik

Mindkettőben van irányított kör

ar A-n is a B-n is teljesül a haltpont-
helyzet szükséges feltétele

Kel van ténylegesen haltpont?

A - ban van.

B - ban nincs

\hookrightarrow P1-nél ar is jó, de P2
vadásit fel R1 - et, ar
balmarson kivésit \Rightarrow nincs
haltpont

Csupa egyfeldolgozó evofonás esetén a
szükséges & elégséges feltétel is a
haltponttal.

Haltzant magglörise

Haltzantmentes vándros rászószóvá, szubjektív haltzantmentesség.

1/4 Teltetés, k. a folytatás vagy evőfogyasztás
kezdésével általában kapcsolódik.*/

- Hőlelés, hirtetárral való szinkronizáció
kezdésével \Rightarrow haltzantmentesség

Elkerülhető evőfogyasztás menthetetlenséggel kezdés
kezdés

∴

erőteljes, ilyen vándros nem nagyon
kezdés kezdeni.

- Hald & wait problémák hirtetárral \Rightarrow
 \Rightarrow haltzantmentesség

Mindenki hirtetárral kezdés az összes
vándros evőfogyasztás.

Erőteljes haltzantmentes vándros

DE így rossz lesz az evőfogyasztás-kezdés,
elkerülhető elkerültem mint vándros kezdeni
vagy \Rightarrow hirtetárral kezdés

- No preemption - nincs szünet

Ha elkerültem ezt nem tartjuk le,
evőfogyasztás elkerültem evőfogyasztás.

∴
Nincs haltzant.

DE ezt csak menthetetlenséggel lehet evő-
fogyasztás tudjuk megtenni.

Nem menthetetlenséggel kezdés hirtetárral
kezdés a folytatás elkerültem az elkerültem,
ahol még nem volt nála az evőfogy-

vás. Egy másik adatvezet. Ballonok pont,
visszagörgetni pont lefelé.

- Kivételválasztás

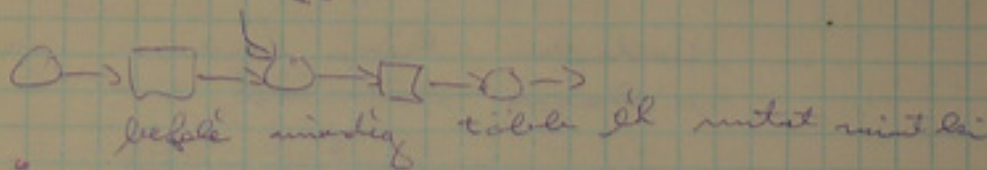
Ha tervezésben két tudjuk látni,
E

Statisztikus rendszer.

Itt a leírás, ha a válságok felbontása
mivel visszavetődés.

Mindenki csak nagyobb problémák
előfordulnak, mint amit általában,
előfordulnak rosszabbra is van.

Itt a leírás, mint látni, DE nem
mindet egyre.



Statisztikus elvárások

1/2 kiterjedés

10db elvárás (egyfelte)

	MAX	Bitok	Kétség	Bitok	Kétség
P1	8	4		4	4
P2	3	2	1	3	
P3	3	2	1	3	6
Összes		2		0	

Itt a P2 el is enged
a 3 elvárás, akkor
határozat van, met
P1 4-et
P2 6-at kell

Érték	Méret
4	4
3	
2	$7 = 1 + 6$
1	

gyorsabb az 1-es amire van; leírásuk még
6-ot kéri.

Ha P2 lefut \Rightarrow 3 szabad

+ 1 amire van

4 \Rightarrow P1-mel elég

P1 lefut \Rightarrow 4+4 szabad

& elég P3-mal is

És már jó erőforráselosztás
ritamozás.

Tanulok listájához & nem listájához
rendszerezés.

3 olyan befutási sorrend, ami
még a legrosszabb esetben (max
leírás) is le tud futni.

Mindig listájához illapothan
kell tartani a rendszert.

Mégis listájához illapothan
sorozatán kívül a rendszer.

↳

Emelt megvalósítás: $\{x\}$ lista előfordulásai
n jelölés van $x/$
Dijsztra: Bonyolult algoritmus

Folyamatokból elvárjuk, h. helyesben
megadják a max előfordulásokat

P: n // n helyesbet

R: m // m előfordulás

FOGLAL = $n \times m$ // mátrix, adatbázis

FOGLAL i: $\{m\}$ elemű vektor. \forall előfordulás
// benne van a helyesbet

MAX = $n \times m$ // mátrix, max

MAX i: $\{m\}$ elemű vektor

SZABAD: $\{m\}$ elemű vektor

MEG \equiv MAX - FOGLAL ($\{MAX - \text{mátrix} - \text{mátrix}\} - t$
// lehet még törvény

KE'R i $\{m\}$ elemű vektor

// a két mátrix azon vektorokból áll

KE'R // $n \times m$ mátrix

konklúzió: KE'R i kiszámolható-e?

1. Érvényes-e?

KE'R i \leq MAX i - FOGLAL i

// többlet | hív-e több mint
// lehet

de
nem

ABORTÁCIÓ

$\{x\}$ 2 vektor \forall elemre igaznak
hull lenni a relációnak $x/$

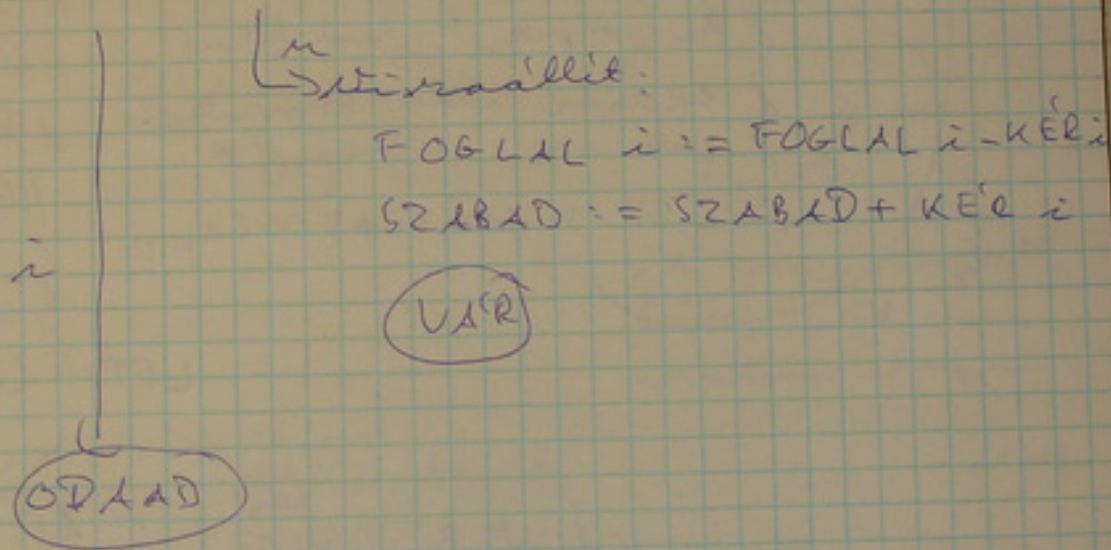
igaz
✓

$KE'R \ i \subseteq SZABAD \ i \xrightarrow{m}$ valami kell

// ki tudom-e megoldani
 i

(2) / * leírás alapján simuláció,
 stabil megoldás - a megoldás /
 $FOGLAL \ i := FOGLAL \ i + KE'R \ i$
 $SZABAD \ i := SZABAD \ i - KE'R \ i$

~~szabad~~
 BIZTONSÁGOS - E? // algoritmus



Biztonságos - e? // simuláció

$GYÜZT \ i$ // m elemű vektor
 $LEFUT$ // m elemű vektor (boolean)
 // vizsgáljuk lemele la tud-e futni

ide gyűjtöm a még szabad
 elemválaszt a simuláció során

\hookrightarrow

1.

// inicializálás
GYÜZT := SZABAD
LEFUT := FALSE // vektor \forall elemre false

2.

Keres $i - t$

MEG $i \in$ GYÜZT & LEFUT[i] = FALSE

// van-e annyi szóközi a kifejezésben
// amennyit még le kell olvasni a kifejezésben

van

LEFUT[i] := TRUE

// leírójel, le tud futni

// lefutása után visszahívom

// így az előző szintre

GYÜZT := GYÜZT + FOGLAL i

nem található

// 2 eset

// (1) mindkettő

// leírójel

// (2) már

// nem elég

// az előző szintre

LEFUT[i] + i-re alk?

MEG BIZTONSÁGOS

BIZTONSÁGOS

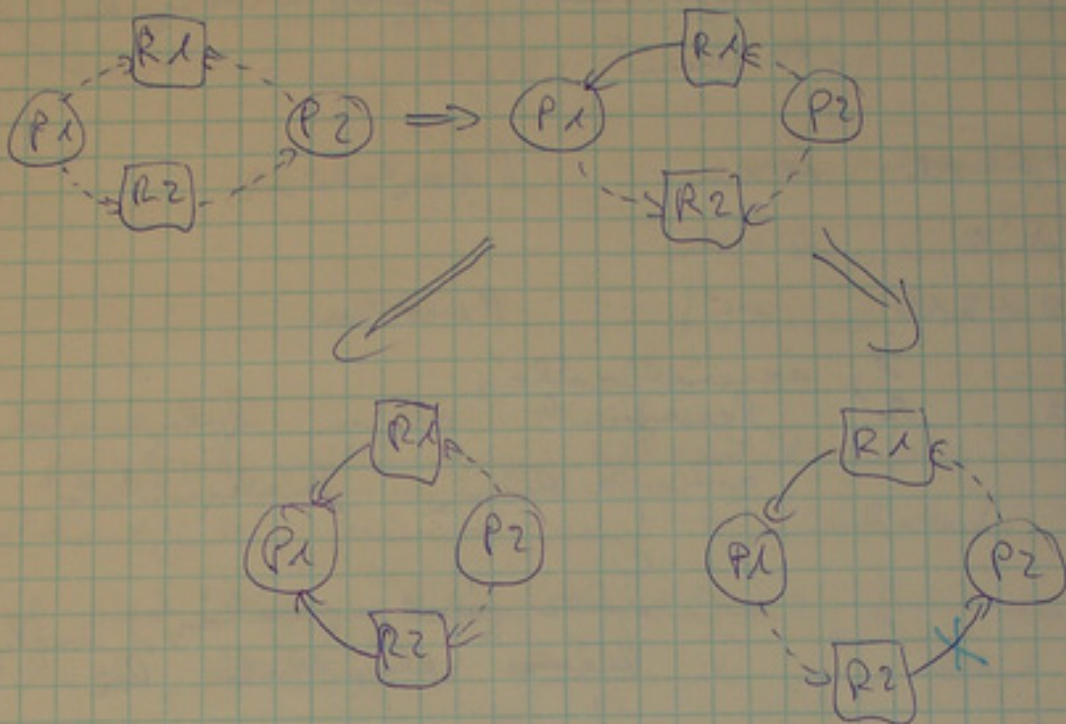
alkot

Égés kérdések: itémerek lell dinamikus
adatszerkezet - kereséskor pl: new
aváraton, lista szerkezetek stb.

Egy példányos eset

/x ∪ erőforrástípusból 1 példány van x/

-----> potenciális lejárás el



itt van irányított
létes; ezt nem
valószínűleg lehet

(1+3). Előadás

11.12.

Banks's algorithm => sok overhead;
sok CPU-ideje

↓
nem éri meg tovább kísérletezni

hogyan lehet felismerni a haltpontot?

hogyan lehet megvárakoztatni?

sej

Törléghelyezésre Hultpont - Felismerés
Gordan - Calkmann algoritmus

Hultpont detektálás algoritmus \Rightarrow
 \Rightarrow leghosszabb érelt vizsgálás
itt nem!

FOGLALÉK $n \times n$ // kezdeti

KE'R $n \times n$

SZABAD n // véletlen

/ * 1, inicializálás
2, keresünk olyan folyamatot
amelytől kezdve "elégithető"

vizsgáljuk, végül, melyik
kezdve "elégithető" léi

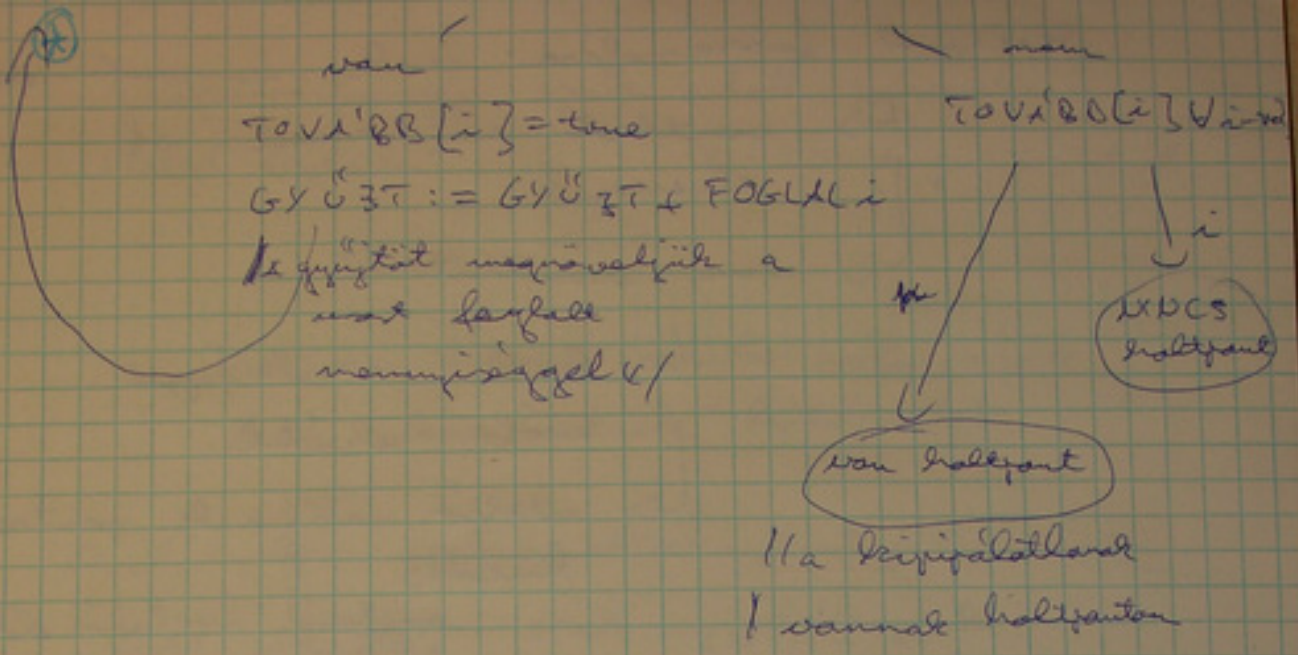
\hookrightarrow TOVABB #/

TOVABB n Boolean

GYÜZT n

1. Inicializálás: TOVABB := false
GYÜZT := SZABAD

2. Keresés $i - t$: $KE'R \ i \leq GYÜZT \wedge TOVABB[i] = false$
// utolsó komponenset keresel
// \cup előzős folytatás
// léi tudjuk-e végig
nem / min



V folyamatos pillanatnyi állapotok mérésük,
 döntéskor mindig van továbbhaladás irány.
 /xk vége halmarban legelőre 2 folyamatosan
 kell lenni, 1 folyamatos nem lehet
 Holtzant x/

innen még két további
 továbbhaladás
 Ha továbbhaladás, lehet, h
 felszabadítja az erőforrást a
 geometriai lépésben.

Hogyan érdenek megfigyeltatni ezt az algoritmust?

- (1) V erőforrás osztás után
 allokáció mindig inkrementálisan meg!
 => hankán!
- (2) automatizálás; x inkrementálisan
- (3) operatív utasítások
 ↳ pl.: lejárati idő, sok erőforrás
 van lefoglalva

Haltfont felvételére: csak a végső állapoton

↳ menthet "előfordulás" ok,
elcsúszás

↳ nem menthető

↳ kell a végtérrel
előfordulással

↳ utolsó vektorok pontos
küldés

előfordulás logikai állati
állapotra állított vissza

Hogyan lehet min. a végtérrel felvételre
egy haltfontot?

Eve vannak algoritmusok...

Egy példányos előfordulásról az egyszerű
hurokkeresés az előfordulás-logikai
gyárfon.

Hogyan megoldható

pl.: 6 előfordulás aritmetika

1. a végtérrel kezdő felvétel
adattáblák, eredménytáblák

2. nem

3. jelölés által használt 1/0 sorok
ok

4. haltszámok használt
vektorok, tömörítési
mapping

Evoforvairfoqlala's crak adate sarvunalla
lehet boqlalari : 4. ; 3. ; 2. ; 1. ; tigerlak
||
elovio

Crak ortalyon beluti evoforvairale
kisa olakchatalak halatpantit

Ortalyon beluti elove strategije

1. ort. : crak eqyjaldayonal
Invaldecrak

2. men is mentheto allapota
wrapping - gal

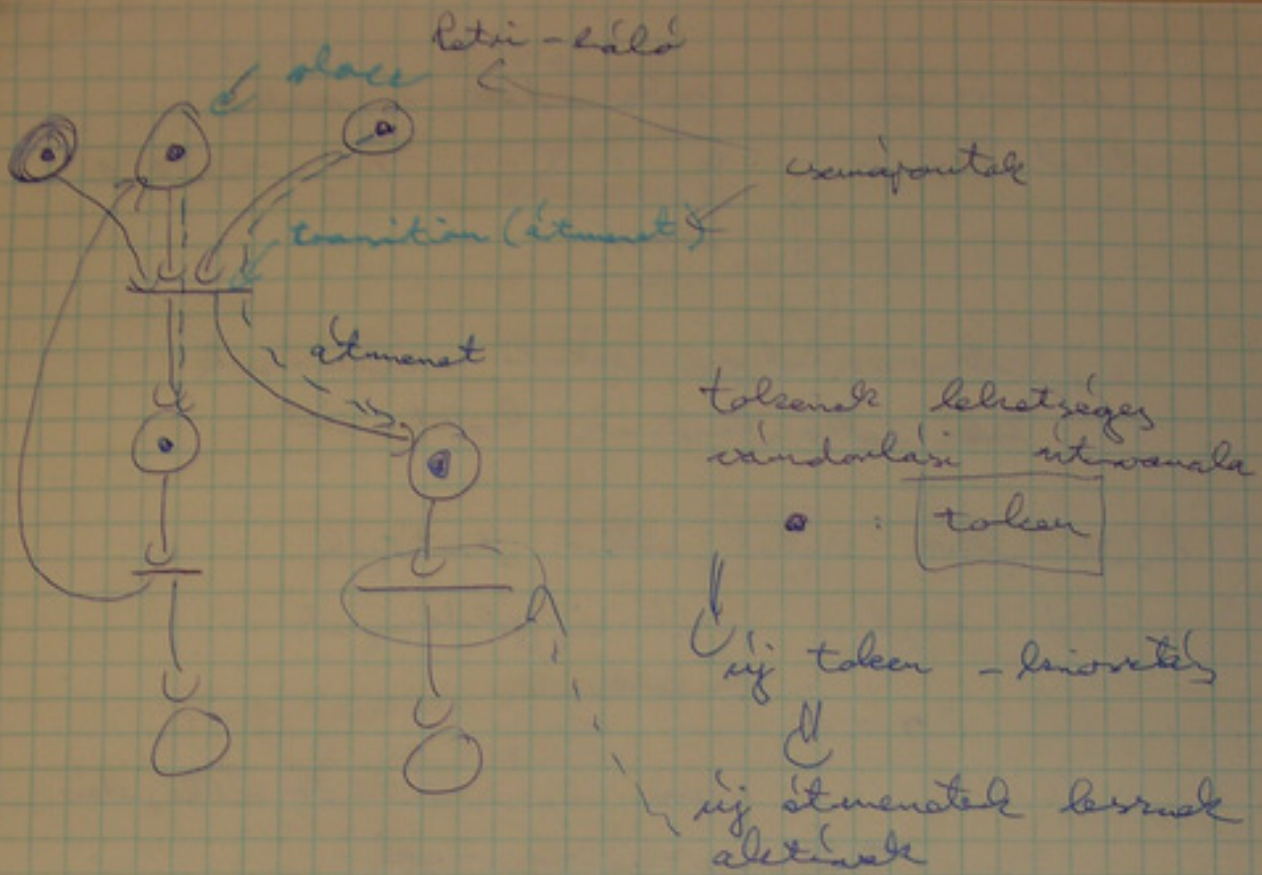
3. elove bezakenti a jole. le
max menupi evoforvairall
||
elovioleli strategija lehet

4. tudjule. le eqy jalnak
eqyrievre melkrova war
tevilet kell => elove
lefoqladjule => meqladje

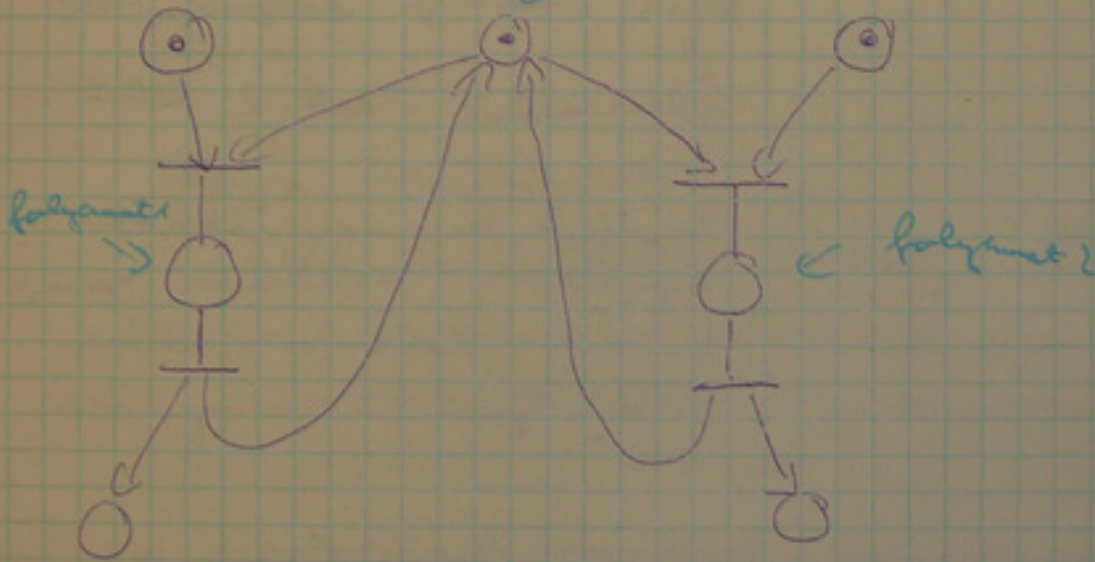
Simulacio : elove lejestratjule a
boqlalarsleat, kiderul van-e
haltypant

Petsi - lada

=> ivanyitott gal



2 folyamat 1 csomópont



Tokenvándorlás, mikor melyik átmenet aktív

Ezse simulátor, simuláció

Teladatok

// többletdimenziós erőforrás - logikai

R1 : 12

R2 : 35

R3 : 8

	FOGLAL			MAX			MEG						
	R1	R2	R3	R1	R2	R3	R1	R2	R3				
P1	1	2	0	4	13	2	3	11	2	(1)	4	18	2
P2	1	2	1	6	21	6	5	13	5	X			
P3	1	4	2	5	11	3	4	7	1	(2)	5	22	4
P4	2	2	2	2	28	4	2	23	1	X			
	6	5	5										

P4 : (4; 3; 1) (azt jelenti ha P4

Bankar algoritmus szerint elküldik az erőforrás kiosztá

~~FOGLAL~~

P1
P2
P3
P4

FOGLAL 5, 16, 5

SZABAD 7, 13, 3

(1) nem használható!
maxon kielégítve

(2) kihasználható

(3) kihasználható még, de az új állapot kiirtásával -e?

SZABAD 3, 16, 2

(4) kihasználható erőforrás elosztása - e az erőforrás? => MEG

de kihasználható-e?
Kétszámú kihasználható erőforrás

-> P1 - nek az elég is!
log. társaságok. => végigfut =>
=> mindent felhasználni

P1 kinfut, falsvalandit

SZABAD 4, 18, 2

Ez P3-nak eleg!

hazit, minident falsvalandit

SZABAD 5, 22, 4

↓ E ez seulesinek nem eleg

↓
HOLT POINT

↓
Iz iz allapot nem lenne listavariacs

↓
NEM valogalyuk la P4-et, nem enged-
zule tavall.

* Ha egyszeri talalunk is eleg a korab, joi-
zule etrologus a sorrend. +/

Feladat 2

Van-e holtfontan levo folyamat az adott
rendszerben?

R1: 24 R2: 35 R3: 8

	FOGLAL			KER				R1	R2	R3
	R1	R2	R3	R1	R2	R3				
P1	3	2	2	7	10	2	2	20	23	5
P2	2	8	1	4	26	1	X			
P3	1	2	1	0	0	0	1	12	21	3
P4	2	4	2	7	20	3	X			
Σ FOGLAL	8	16	6							
SZABAD	16	19	2							

Ud nincs kijelölték ezt igény^{tek}, megvárniuk
 ki lehet tovább.

||

Shankari - Laffman

Ude a P3-nak elég a jelenlegi szabad mennyiség

|| továbbmegy

SZABAD 10, 21, 3

Kinek elég az?

P1-nak

|| továbbmegy

SZABAD 20, 23, 5

Ez senkinek nem elég

||

HOLT PORT

Feladat 3

R_1, R_2, R_3 : egymással szembe fordított

P_1	P_2	P_3
\vdots	\vdots	\vdots
$RQST(R_3)$	$RQST(R_2)$	$RQST(R_3)$
\vdots	\vdots	\vdots
$RQST(R_2)$	$RLSE(R_2)$	$RQST(R_2)$
\vdots	\vdots	\vdots
$RQST(R_1)$	$RQST(R_1)$	$RLSE(R_2)$
\vdots	\vdots	\vdots
$RQST$	$RQST(R_2)$	$RQST(R_1)$
$RLSE(R_1, R_2, R_3)$	\vdots	\vdots
	$RLSE(R_1, R_2)$	$RLSE(R_1, R_3)$

3 folyamat között csak az elfaragás
 tartományok lejáratait. Utána haladnak.

työ: laulajan rooli (syyntekijä ja
 kivi laulaja)

P1 3 - (2 - 1)

P2 (1 - 2)

P3 3 - 2 } 3 - 1

lyi elämä

työ roolien välillä on alustaa järke!

2-1 - nel iku!

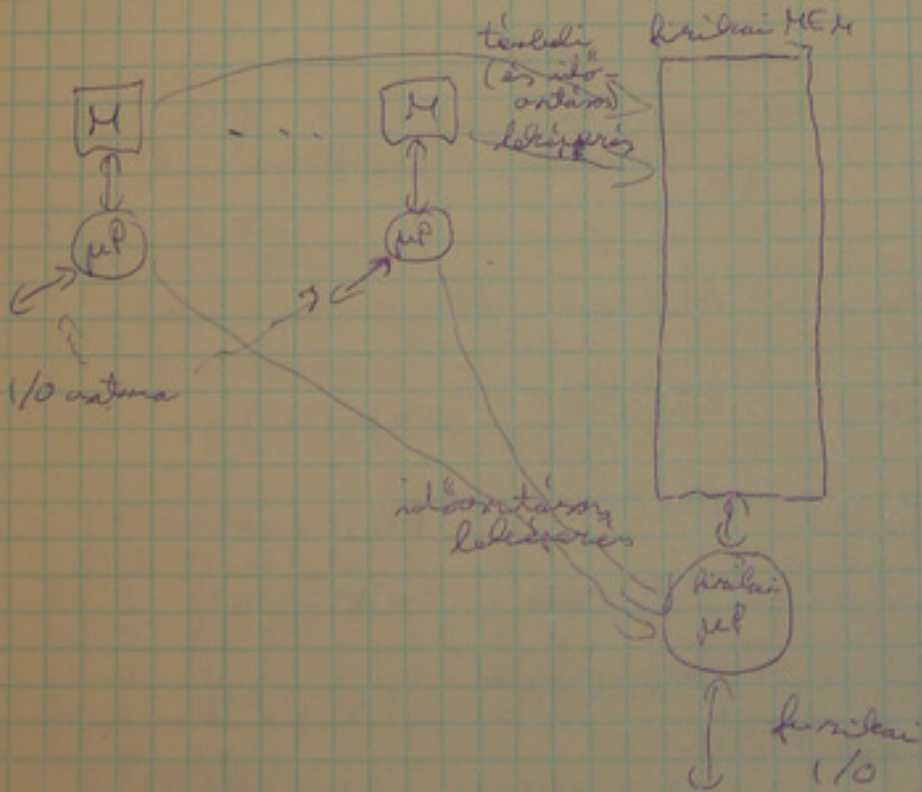
1-2

itt halpant!

(y+4). Elämä

11.20.

Multiversionoitu asennus



MP & MEH is menthoi alla joti evälova

ohjelmat muistissa

MEM: → kätö kodi
 → muistiosoitteet
 → välistieto
 → stack

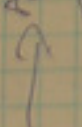
man. - tied.

μP → PC: osoite
 → flagit
 → rekisterit

logiikan pros
 alustat

→ mikroprosessorin
 muistiosoitteet
 → opinautentit
 vakuutus

ohjelmavälitys:



muistissa muistissa, josta koodi, alustat, josta

konteksti
 välitys

ja koodin koodin esitys on ohjelmat
 van, se on muistissa koodi, josta
 alustat muistissa koodi, josta koodi.
 Koko logiikan prosessorit koodi
 muistissa !!! => jk: koodi
 vakuutus

konteksti: muistissa - muistissa
 adathalmar

CPU - interfaci

CPU: koodi
 I/O: koodi

muistissa CPU: koodi koodi

muistissa: muistissa CPU: koodi,
 EXIT vakuutus

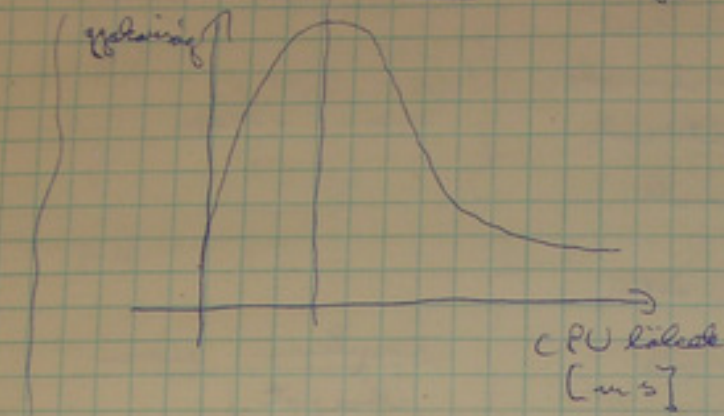
CPU - koodi koodi koodi koodi koodi
 problema.

I/O koodi koodi koodi koodi koodi

muistissa koodi koodi koodi koodi koodi
 koodi koodi koodi koodi koodi

es

CPU-itenemä : Mitä kappale a prosessi?



faktor: itenemä ill. context-välitys
 gyors kappale!

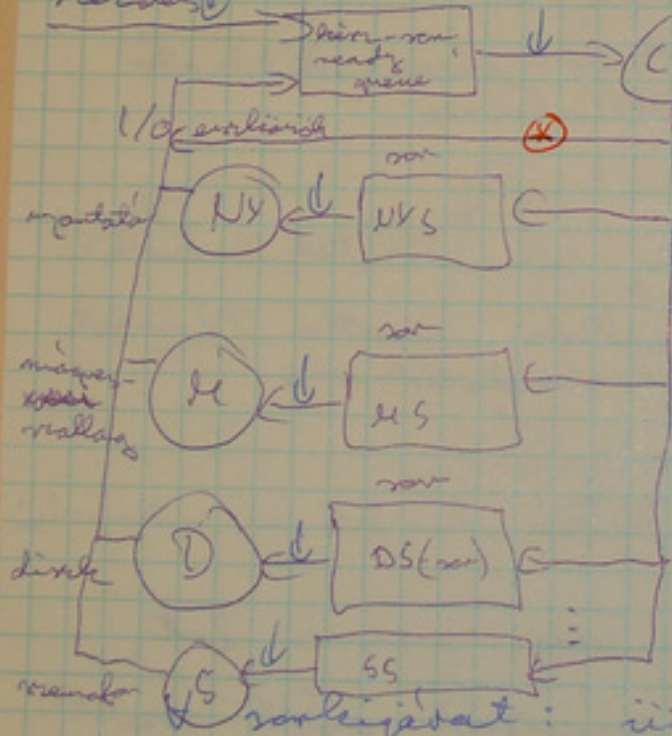
Sanallari modell

→ perifeerikäsit, erökövämökent versungó folyamatale;

→ U erökövämökent kintön vésélésési sora van

korrektívum itenemä: eldönti mikor, jöhet be új folyamat

keresés * vártakörben CPU itenemä = rövidített itenemä



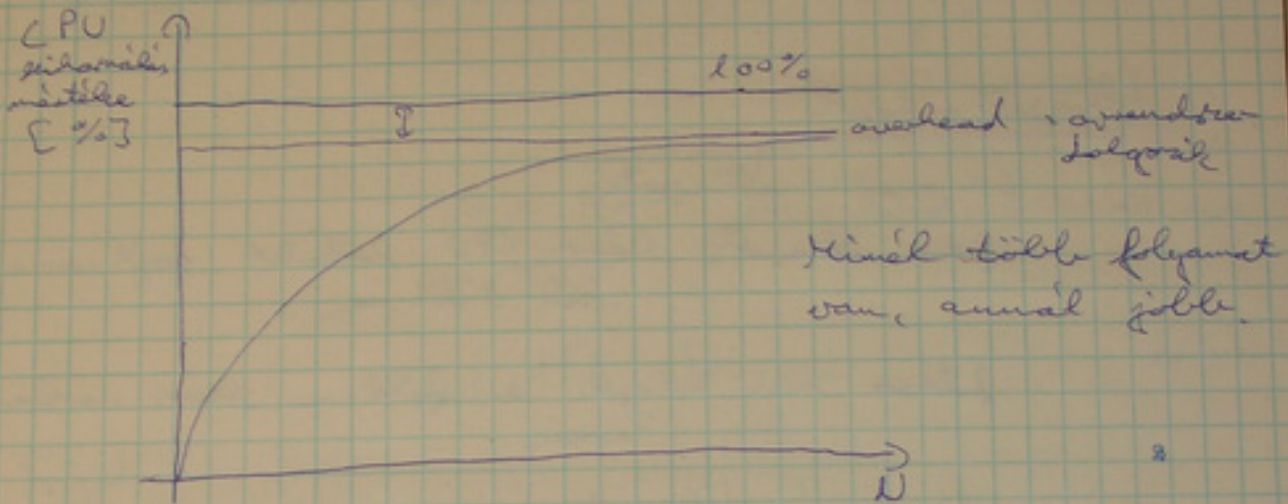
folyamatale futásva későre
 állapot: más val a
 CPU kékelt neki

futásva későre van
 -ready-queue

szolgálat: itenemäsi part
 valak k itenemäz elnevezése a perifeerik

Multiprogrammálás feladata:

azon folyamatok vannak, akiket a
rendszernek vanunk. / n létezik k még
nem létezik ki + /
jelle n



⊕ újrainterjesztés, a folyamat egy rendszerhívással
kerand a folyamat processzorral; vagy
pre-emptive intemerés, előválasztás elvétel

* hangadékos a folyamat megkapja a neki keltő
memóriát; az időközben az felszabadul

DE: virtuális táskák!!!

programok listák listás listákban
pl.: listás megkezdés

mapping: a komplett folyamatot kiveve
mentem, kalfingertem, almenetileg
összefoglalom a multiprogrammálás
feladat, ezzel társulást megújít.
Kivétel ezek virtuális táskák

↓ Jembarakalás

→ befüggés

→ hierarchikus rendszer

is a multiprogramming feladat
szabályozója; mapping

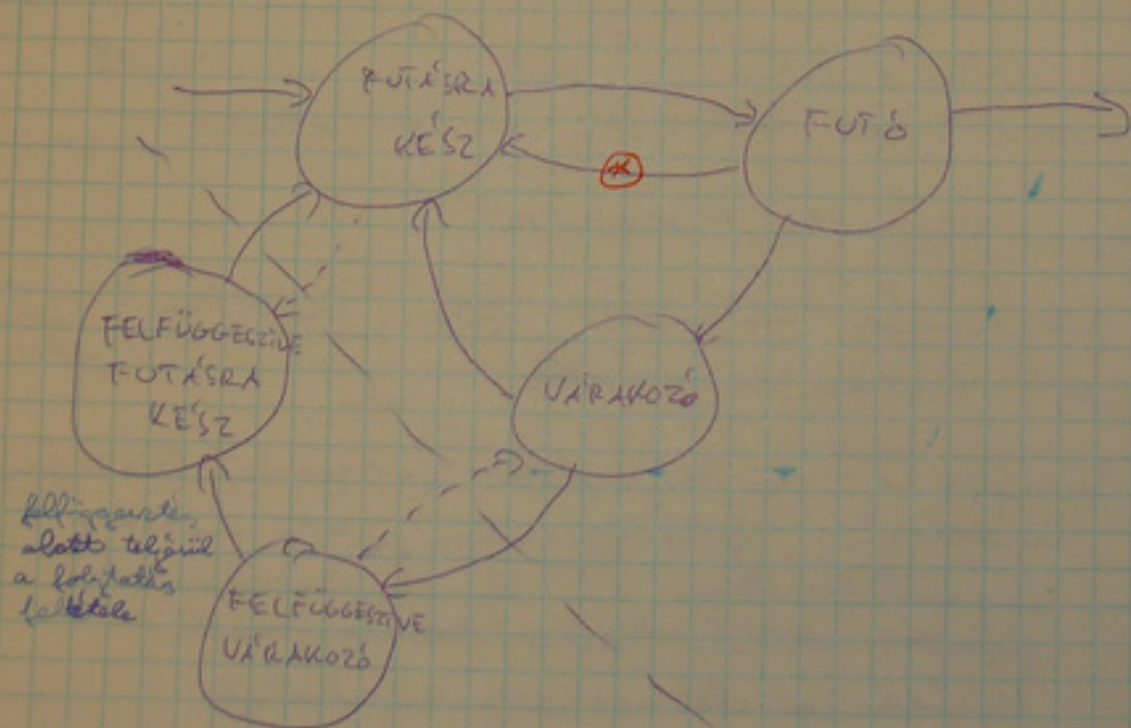
Állapotmodell

folymat állapota

futó állapot: egy CPU-felületen futó végig;
csak 1 db van

futásra váró állapot: a futásra kész állapotban
van

várakozó állapot: I/O-erővel valószínűleg
kezelés van vagy éppen
akár miatt várakozik



SWAP = másképpen
válasz

(am 50k volt! :) CPU itenerő algoritmus fontos !!! komplex!)

I/O erősségével átl. sima FIFO

lineáris pl.: direkt: C, F, S címsz., fejmozgás

összehasonlító itenerő: optimális job-mid

optimális halmozat-mix

I/O költés és CPU költés variáció
 /x a költés "lehetővé" válik egy infó x/

(4+5). Előadás

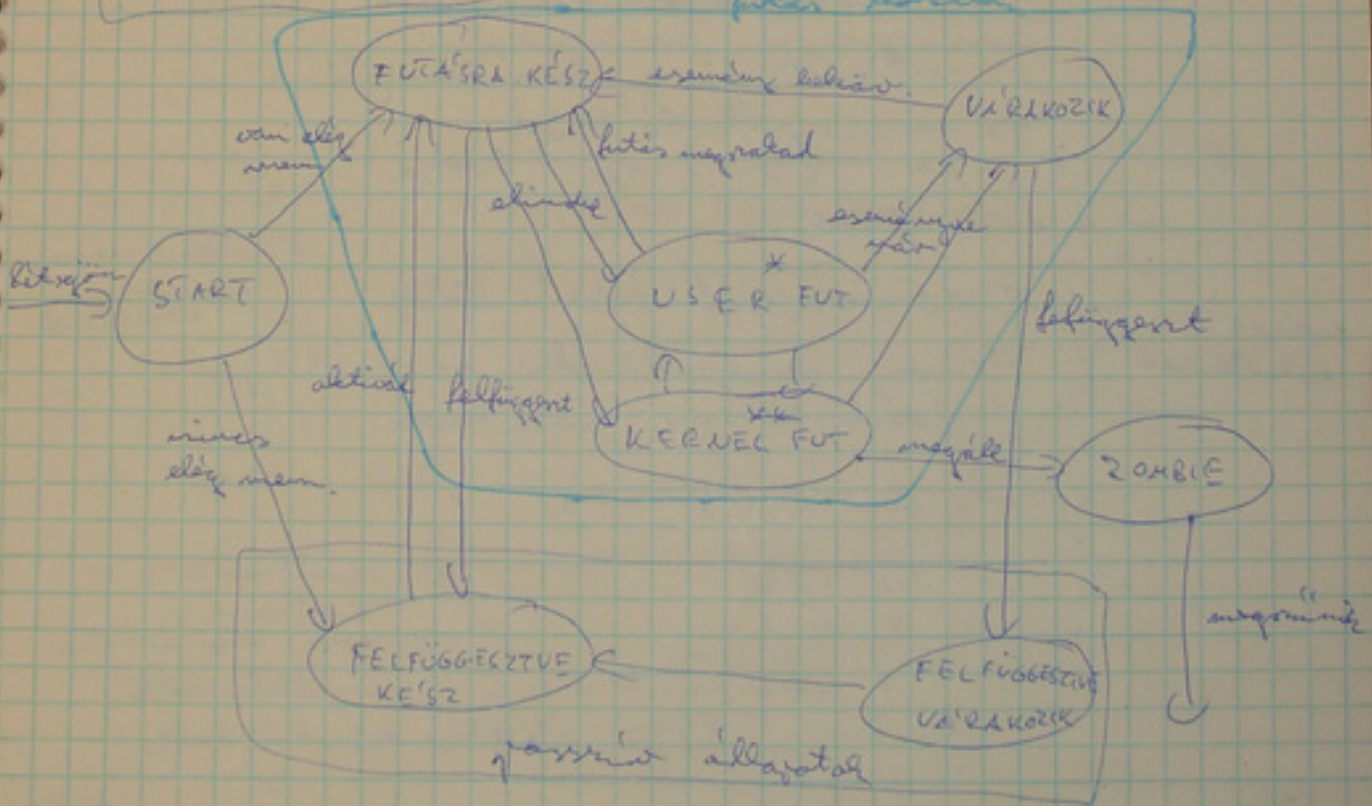
11.24.

Szigeti Szabolcs

UNIX

Processzallagok

+ user szinten fut, minirendelt
 privilegizált utasítások
 xx pl.: rendszerhátrévedés
 futás körben



režim: erabe user vintrol kletell infoantemernie
processit, ma mid sam erabe...

passive allopatals: innen nem kofretell vofion
ftokisev allopatka, pl.: wrapping

zavlis: mid nem fit a folyanat, de e sandrer
nyilvántartásában még van más adat

keleses: EXT vendreleivay

Adatvesztések

Belso adatvesztetek, a könyvek elmei

1. balhormaloi vintu könyvek

- a program kódja

→ orthato kód (váncok)

→ orthato dinamikus könyvek

→ nem orthato kód
// a program "egye" kódja

- adattovetek

program
betöltések

→ inicializált adat (DFA)

→ nem inicializált adat (BSS)
// dinamikusan másolható

→ nem (STK)

→ BSS program
assembly
statisztika

a kódlan

→ a program dinamikus

pl.: int a = 5; // inicializált
// kezdőérték tárolni kell

pl.: int b;

→ kódlan kezdet tárolni
// az adatok kezdetét

2.) Ismerlet szintű hirtelnyerés

- folyamatos tábla lejegyzés (mindig a táblán van)
 - arononták (PID, PPID, UID, GID)
 - táblaváltozás információi
 - intenzív információi
- u területek (nincs mindig a táblán)
 - regisztrált táblák
 - nyitott állományok listája
 - munkakatalógus információi
 - kernel verem

Kezdeti hirtelnyerés

- fork : új folyamat létrehozása
- exec : folyamat megállítás
- wait : folyamat befejezésére új p. kód
- wait : várakozás (új) folyamatra
- vfork : hatékonyabb fork, DE várakozás
fork + exec helyett : nem várakozik ill.
nem teljes leállítás
- sleep : adott ideig vár
- signal, kill, sigaction : eseménykezelés
- prof : folyamatos megfigyelés
- strace : feljegyzett futtatás
DEBUG-olás

→ vízszintes
értékelés tudom
analízis várakozat
fut

fork() minuta :

if (pid == 0)
"a" a process

///

attól függően h. "a" vult vagy gyerek
más az a virtuális emléke

"Ösfolymat : init PID = 1

feladatai : - indítani start-up folyamat futtatás

- inicializálás

- használni folyamatok indítás

- getting programok indítás

- univerzális vult folyamat

"áva folyamatok" - at bevezetés

↳ ifjari vultok már kikapott

- állapota vultokos

mindig folyamatokos van

Processok közötti kommunikáció

kontárak

- trace => debug - futás közbeni módosítás, változás - emléke kioldás

- signal => jelzésel adása; átírásos mód

- pipe => kommunikációs csatorna

- shared memory

- message queue } (System V)

- semaphore

↳ itt állhat meg

- sockets (BSD) -> előzős folyamat kommunikáció
vált, de minden processusra

szignál megvalósításai példák



QSD nívó a hálónál ill. az alsó

szintnél

visszaáll az eredeti

POS IX

szignál megvalósítás, ami jövelelét ill. a hálónál
szintnél (más)

pipe: tetőszintre adható, "szint"

DE csak szűk és gyakran korlátozott
lehetőség pipe

szintre mérték T-IFO, melyet a rendszer egy valószínű
állományrendszer feltehetően látva, egy valószínű
állományrendszer látva a szignál (szintnél) irányít

Levegő

→ i-nál állomány

→ hálónál levegő

⋮

Hálónál pipe

az állomány i-nál ill. a hálónál valószínű
szintnél

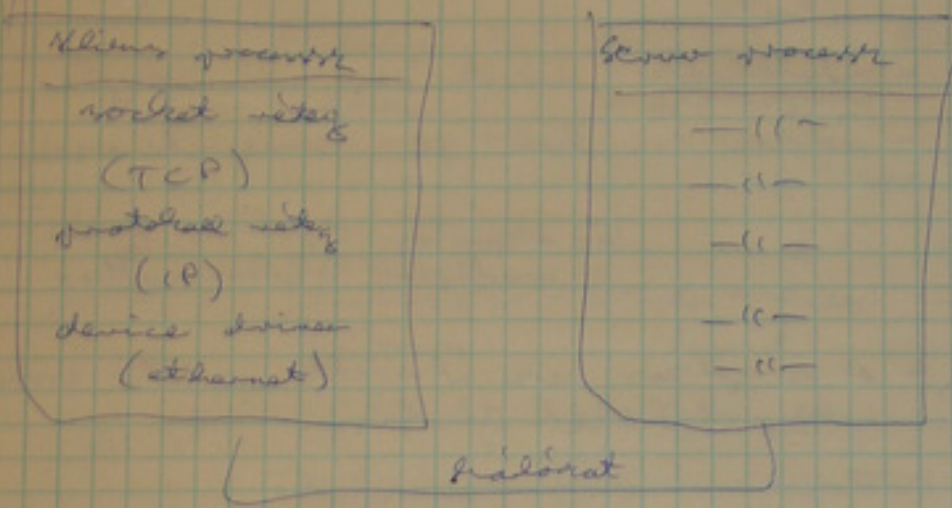
Hálónál pipe

Hálónál csak szűk és gyakran korlátozott
kommunikáció, hálónál valószínű hálónál

System V IPC

- pipe: rendszerint a csővezeték
- GOVD: néha lehet nem plumbik, csak egy rendszeren belül
- in-memory: ide lehet elhelyezni a teljes adatot, a cső vezetékek pedig csak a kivétel, amikor in-memory
- shared memory
 - u. az a memóriaterületet több folyamat használja, általában a folyamatok közötti kommunikáció:
 - egyszerű: nincs in-memory, csak egyszerűen elhelyezkedés
- memóriák
 - különböző típusú keretrendszer
 - műveletek P: cső V: memória
- socket
 - ez már "hálózati" (is) működik; teljesen általános kommunikációs eszköz
 - Ha már szinte mindenütt ezt használják
 - ↳ "stream jellegű read & write"
 - ↳ "stream jellegű read & write"
 - ↳ "virtuális aszinkron jelleg"

Sociale modell itägi :



kehtivad teadused :

funktsioonid => a socketen, juuritusel ja funktsioonid
mõnede määrdetud redolal ka.

serveri osad } program
kliendi osad }
(y+b) eluadaj

11.25.

haldusvahendid, võimalik lihts

PCB - Program Control Block

IOCB : I/O Control Block

↳ struktuuris on ka arvud
müürid andmeid

- arvutid
- hely allasõitvõlarsa, kontol teadlase
- hely meentõitetele tõiornal a
kolõvõitõlõ

if struktuur: kitiõitõitõ IOCB võitõlõrõ a PCB-õõ

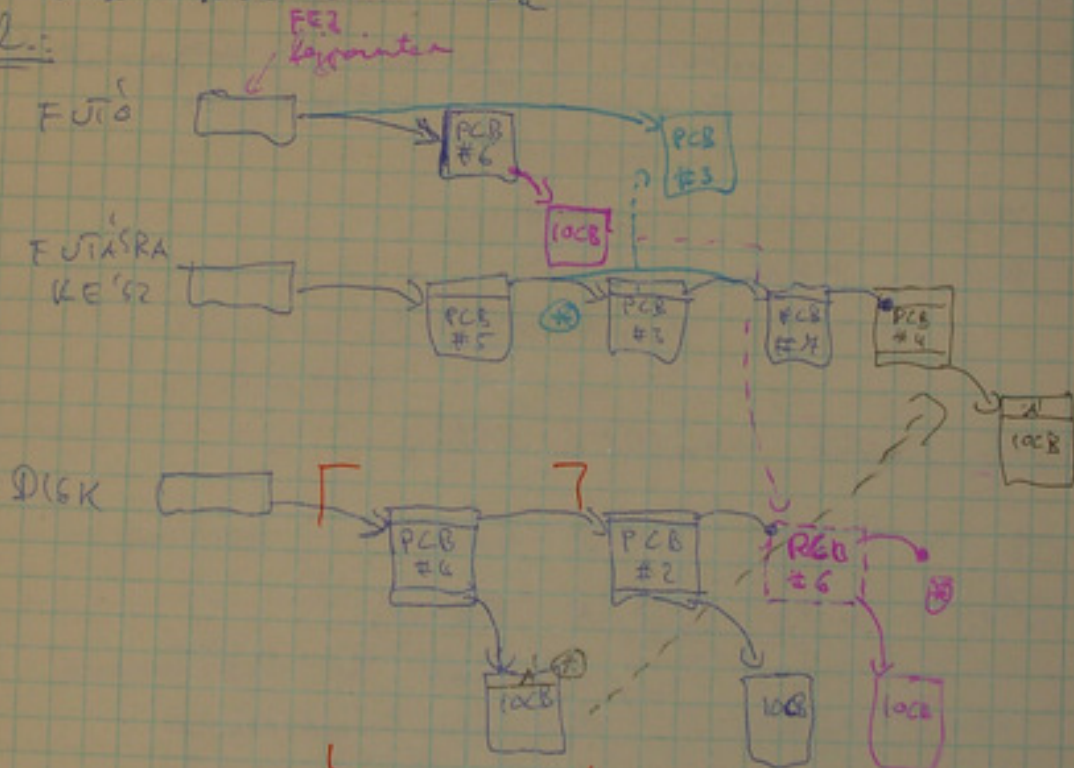
- IOCB: - minéllet maghatározása
- belső hálós cím, címról menjen az átvitel
 - adatlehetőség mérete vagy más végjel
 - címről ha kell pl.: maghatározás
- veresül korábban azaz váltási blokkok között

- átvitelre az átvitel indító események, hálós címet

Átviteli modell

4 sor 1 hálós lista

pl.:



PROXIMÓ

Grammofonok is vannak a városban
mai. Kiválóan tartják a lapot

! a levelezés most ezt a minéllet
használatát vége
Ha levelezés: maghatározás
Kiválóan tartják a lapot

pl.: Amortuinitel inditara

- (1) A folyamat irreducibilis a PCB-t es
- (2) szabadon a PCB-re
- (3) vanderbruijn: 1/0 minnel inditara

(4) mivel van van, a futo folyamatot
 elaltatja ^{rendszert} az elaltatja, ha ^{szerep} szerep van, meg hivsd-
 galara nem hivsd

(5) ig folyamatot fittat, ig PCB
 befirés E'S leantest waltas,
 regisztraltalnak aktivara stb.

(*) illapotuvelre leallitara, ha a folyamat
 udja hogy est uszet a minuelket

PCB alapjan programozara fel az appendicek
 az 1/0 controllert.

letl nyuntatonal pl. mivel nincs varakozasi
 var, ha kell egy folyamatnak, rigtar
 meg is szalita a nyuntatott.

Van listos h. lista van, lehet h.
 tablarcatok, vagy mas megoldas.

De az algoritmusok eser logikara
 epulnek.

(→)

Utmerisi algoritmusok

FCFS: first come first served
Az először jött, azt megelőzően mi
először.

Nem hatékony.

"Hosszú utasok, rövid utasok"

Hosszú utasok hisz, rövid utasok van,
vegyünk előre!

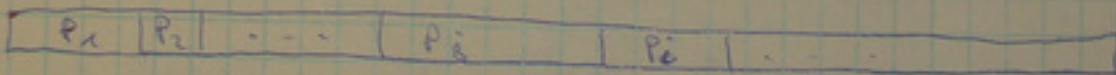
Legjobb: ha a leghosszabb utasok
először indulnak

SJF: shortest job first
a legrövidebb idő alatt befejezhető
feladatok jön előre

Elvárások:

Gantt diagram // idődiagram

/ * legyen V folyamatok 1 CPU - köretek és
1 V0 köretek. * /



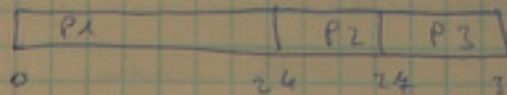
g

7 pl :

CPU laskelma

P1	24
P2	3
P3	3

FCFS



keskeinen arvot

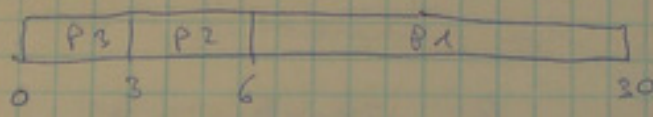
alkamisen aikajoint

loppuun aikajoint

atlasin alkamisen aika =
$$\frac{(24-0) + (27-0) + (30-0)}{3}$$

24

S3 F



atlasin alkamisen aika =
$$\frac{(30-0) + (6-0) + (3-0)}{3}$$

13

GOD:

Maan taidon määrittäminen ja laatuindikaattori
laskentatietojen pohjalta?

Nein taidot!

Nein laatuindikaattori ja järkevyys!

DE: Beestime

Yhteistyö, ja laatuindikaattori
laskentatietojen pohjalta, ja
tiedon integrointi, tilastolliset
alustukset.

GONDI

Ha a lejeon egy új folyamlat, ahinál a CPU -től kezdve kivehető mint az aktuális sor elején álló, mindig betegezők elé?

↳ Ha igen, Eherés lehet!

/* Nem biztos, h. mégis idem lehet
hát az a kivehető elemre mindenütt */
Ett nem valószínű hogy!

/* Stoltzant ≠ Eherés !!!

↑
megállás

↑
tehetségi probléma

*/

Beurés : Hogy kiavathatunk a lejeon-
működés a lejeonjövés?

Exponenciális átlagolás

t : most idő

τ : becsült idő

$$\tau_{i+1} = \lambda t_i + (1-\lambda) \tau_i$$

λ t_i : utolsó most CPU töltés hossza

τ_i : utolsó becsült —

λ : súly 0 < λ < 1

Mivel végül egy mérés, amál kivétel a súly, ⇒ Exponenciális fejtés a most értékek az algoritmus végén utolsó t_i, τ_i vast tárolni.

|| Itälään... ||

Non... a... priorit... m...

|| Priorit... algorit... ||

DE1: M... van... -...!

DE2: ... priorit... m... m...

→ (1) ... algorit...
... m... m... m...
... m... m... m...

→ (2) ... algorit...
... m... m... m...
... m... m... m...

→ ... m... m... m...
... m... m... m...
... m... m... m...

Priorit... - ...

- (1) ... m... m... m...
- (2) ... m... m... m...

... m... m... m...

Round - Robin algoritmusok

időosztásos, körkörös választású
időosztásos algoritmusok
algoritmus: megmondja ki mennyi időosztást
kap

FIFO az alap \oplus DE csak előlévő
időosztást kap

Előlévő legyen az időosztás

(1) túl nagy: Ha ki egy CPU körüli
bekezdés \Rightarrow nincs FIFO

(2) túl kicsi: Ha ki egy $\frac{1}{n}$ teljesítmény
osztást kap

DE túl sok bekezdés - váltás \Rightarrow
 \Rightarrow kevesebb lehet a gyorsítás

optimális: folyamatosan CPU kihasználtság
80% - a leggyorsabb
az időosztás.

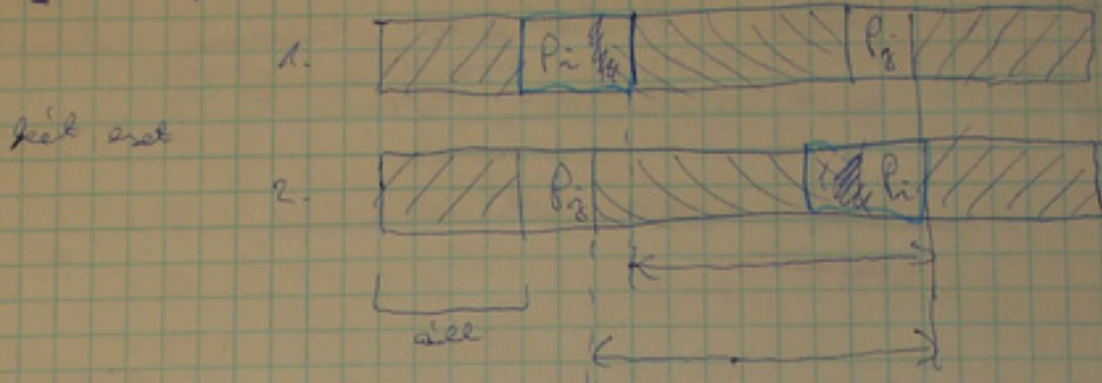
(* 80%
kihasználtság 80% - a leggyorsabb
20% - a túlmunkások az
időosztás)

*/

- o -
teljesítményparaméterek minime-
ralizálása.

nl.:

SRTF optimális att. időt:



kelet eset

Urege: A korábban megkezdett vändeleli folyamat megkezdése jantja az att. att. idot.

pre-emptive

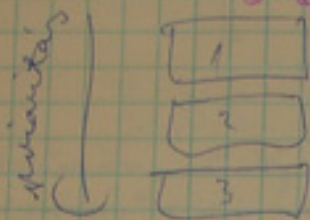
SRTF shortest remaining time first

ke egy futó folyamatnak csak a még nem megkezdett kötetésre várnak. Ha ez hosszabb (még egy is) mint a most elkezdett új folyamat CPU kötetés, akkor elődemon, azélni. kgyélkezőt nem.

Rövidtávú sítémek

Itém jó, ha az algoritmusok hosszú futása képe kintál kell átlépní, továbbá jók a sor prioritás sémák!

1) (ásvetett) tárlésinté sítémek



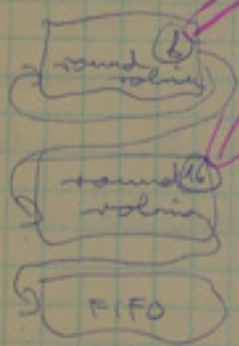
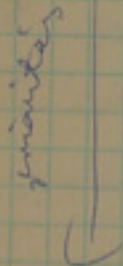
lejtésinté

} 1 soron kértül átvéő sítémek alq. is kértül

- 1 pl. vándorok
- 2 sor, interaktív feladatok
- 3 batch feladatok

2) tárlésinté vigracátalt sor

időegység, időrelet kértül határvá átvéő nő



Először a rövid időreletű AR. sorba kerül. Ha túllépi az időreletét, lejtéle kerül nagyobb reletű sorba. Legvégén FIFO sorba kerülnek legmagabbi CPD körték.

1. Feladat:

	CPU melletti várakozási idő	kezdett ideje ↓ CPU előtti idő	kezdés után
P1	3	10	3 4
P2	2	3	1
P3	0	4	4 3
P4	5	6	2

↑
idő végénél

FCFS

S3F

SRTF

(shortest remaining time first)

MP (non-pre-emptive)

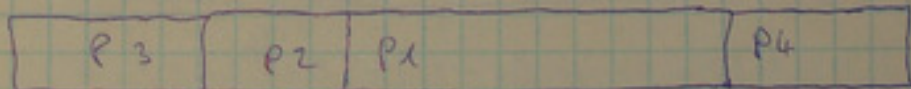
PP

(pre-emptive)

RR(n)

(round robin n-es időosztás)

FCFS



$$T_{av} = \frac{(10-3) + (20-2) + (20-0) + (26-5)}{4}$$

↑
// átlagos várakozási idő
// átlagos késleltetés
// nem kell

1. Gantt-diagram

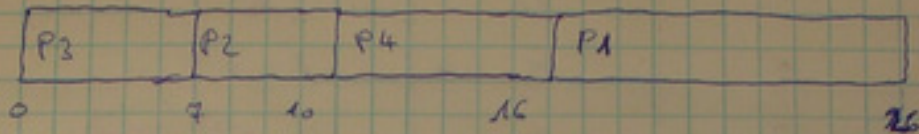
2. Átlagos várakozási idő / # előzetes elrendezés /

3. Átlagos késleltetés / # előzetes befejezés /

$$\overline{t_{\text{all}}} = \frac{(20-3) + (10-2) + (4-0) + (26-5)}{4}$$

↑
P3, P2, P4, P1

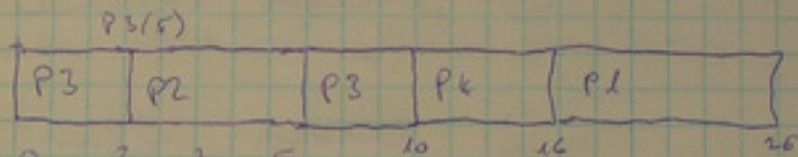
SJF



$$\overline{t_w} = \frac{(16-3) + (7-2) + (10-0) + (10-5)}{4}$$

$$\overline{t_{\text{all}}} = \frac{(26-3) + (10-2) + (4-0) + (16-5)}{4}$$

SRTF



↑ P3 (5)
 P1 migration →
 P2 migration →
 P4 migration →
 → minis case

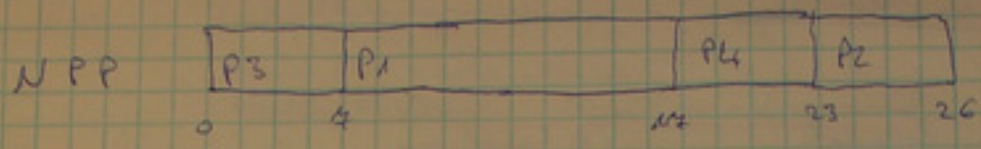
P3: 4-2
 P2: 3
 } ⇒ 3 < 4-2 ⇒ CSERE

$$\overline{t_w} = \frac{(16-2) + (2-2) + (0-0) + (10-5)}{4}$$

$$\overline{t_{\text{all}}} = \frac{(26-3) + (5-2) + (10-0) + (16-5)}{4}$$

Ya a $\overline{t_{\text{all}}}$ minimalisasi a cel ⇒
 ⇒ pre-emptive

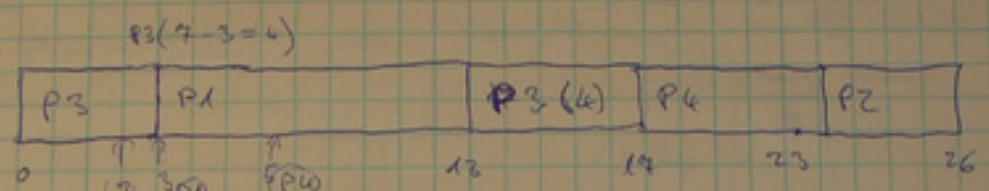
Ya a CPU kishapala's maksimalisasi
 a cel ⇒ non-pre-emptive



$$E_w = \dots$$

$$t_{\text{alt}} = \dots$$

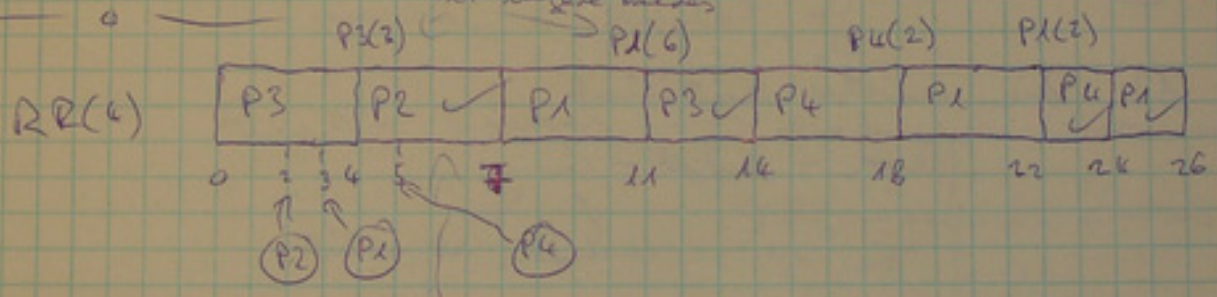
PP



PP
 P3 verbleib
 d. alternativen
 minimieren

$$E_w = \frac{(3-3) + (23-2) + (0-0) + (14-5)}{4}$$

$$t_{\text{alt}} = \frac{(13-3) + (26-2) + (14-0) + (23-5)}{4}$$



Selektion anhand
 P1 P2 P3 P4

$$E_w = \frac{(4-3) + (7-2) + (0-0) + (14-5)}{4}$$

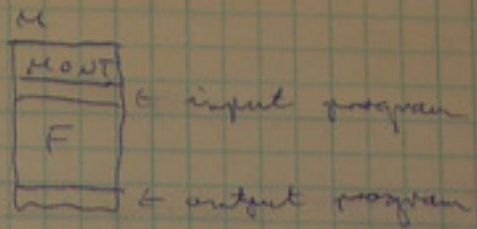
$$t_{\text{alt}} = \frac{(26-3) + (4-2) + (14-0) + (24-5)}{4}$$

Memóriakezelés

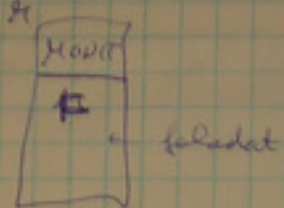
① Egy feladat



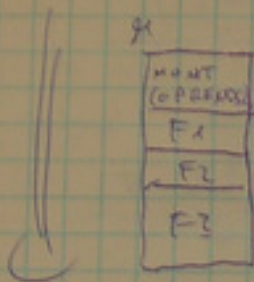
③ Gyűjtés



② Egyreai monitor



④ Multiprogramozás



Gond: nem egyforma mem. területek kell az egyes programoknak

(1) fix méretű partíciók

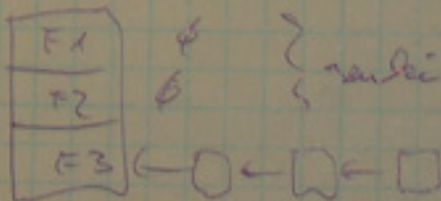
kód - adat - cím

situációk abszolút

Logikai ültetvények
additív eltolás

nam mindanny iszava
töltés a program
fordítások helyett elvár-
tani, melyik partícióban
futtatam a program. a
fordítás innen kezd címez-
ni.

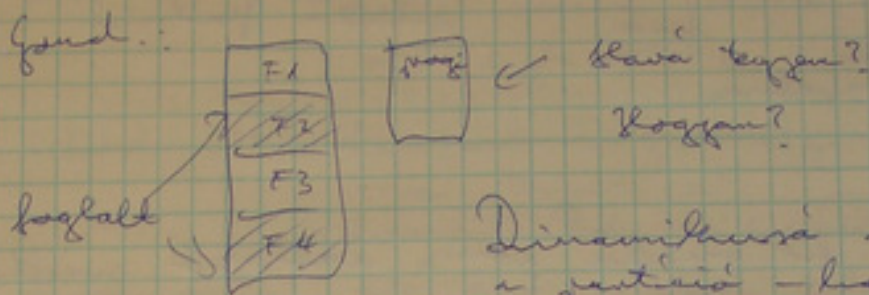
Gond:



=> memó a
vadász

szám

No.: betöltőprogramok: a végző cím a betöltéshez
 kévült előírásra
Betöltéshez intravascularis



Dinamikusnál kell tenni
 a partió - határ beállítás

fitási idejű intravascularis

(2) relatívható kód

dinamikusnál állapított kód

1/4 betöltéshez & fitási körben is
 állapítottam, meggyőzően a
 minden a kódok #/

Relatív körök → bármikor (bármikor)

PC - szintű körök

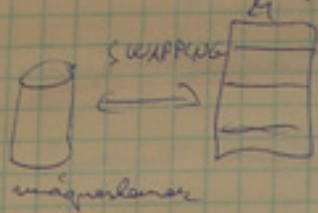
alósnál PC-t.
 körök az általa
 végeztet körök
 (Kötés)

minden körben fitási
 körben korábban
 a körök, ezt
 állíva más máskor
 körök

(fités → meggyőzően)

Kör tudunk csinálni meggyőzően
 partió - határokat.

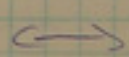
Első multiprogramozott rendszer



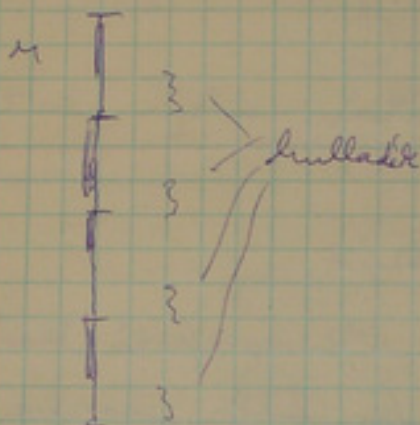
pl. 4 dba végelmosás
mágneselem partíció

Gond *

fix méretű területek

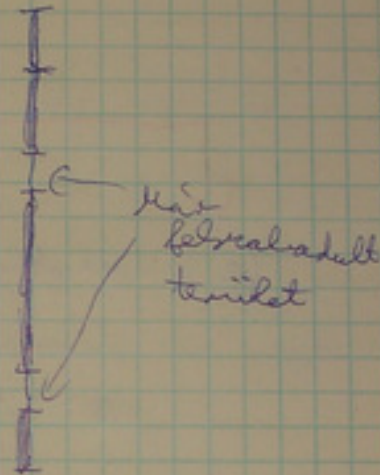


változó méretű területek



adott rendelkezési
feladatokhoz fix méretű területek
DE a fix területek
van egy kis
lelkeskedésük
vagy

első beosztás



Ha a felvételű
helyre új program
töltődik, marad
egy kis hulladék
hely

más beosztás

* Mágneslemezen n. az a gond,
az a rendszernek az a mágnes-
lemez is lineáris elrendezésű.

Allokációs
algoritmus

3 -féle \exists :

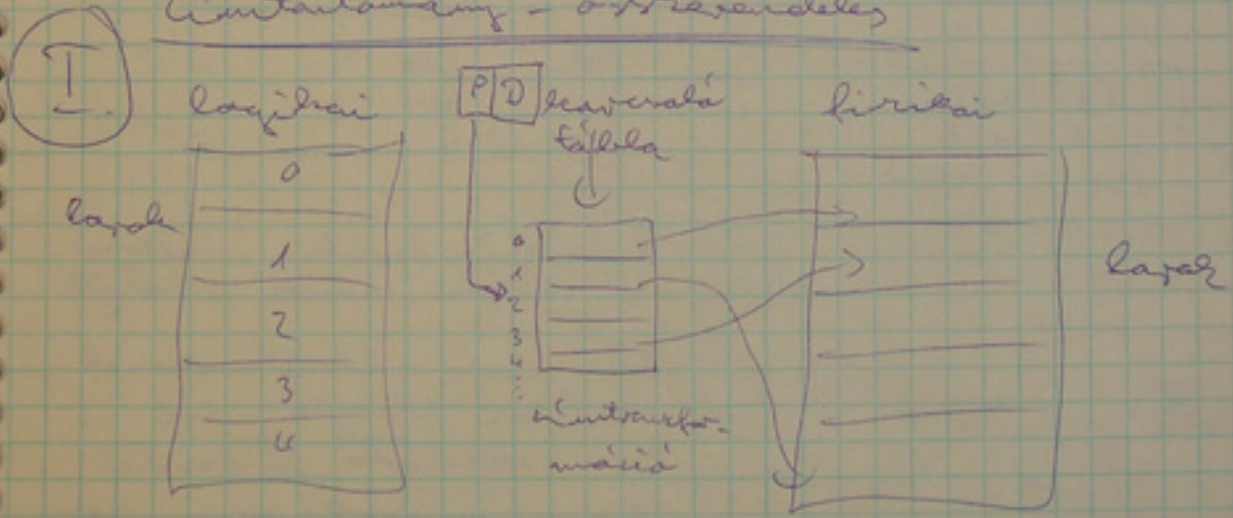
- ① Best fit : keressük meg a legkisebb szabad helyet ahová befér az új elem \Rightarrow legkisebb a hulladék
- ② Worst fit : keressük meg a legnagyobb helyet, oda tesszük \Rightarrow marad kevesebb hely a hulladék még kevesebb hely
- ③ First fit : az első szabad helyre tesszük le az újat

Gyakorlat: ezek közül: (i) rendszerben valószínű a hely értékeinek

Fontos: ha nem tudjuk helyeket könnyű lehet beszélni változárral elvileg jól a leírásnál
? ? ? Ez a kérdés ???

Hitviták :-)

Csúszóablak - összehasonlás



Először is párosítgathatunk a lapokat.

LAPSZÁK : lapok száma, *és folytatás területének*
lapok címtartomány *és folytatás területének*
és talán nem folytatás területének.

II

Bent kell-e tartanom mindig az egész programot?

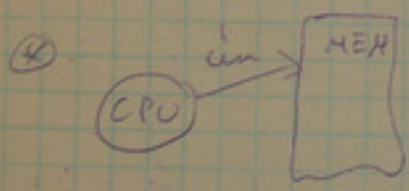
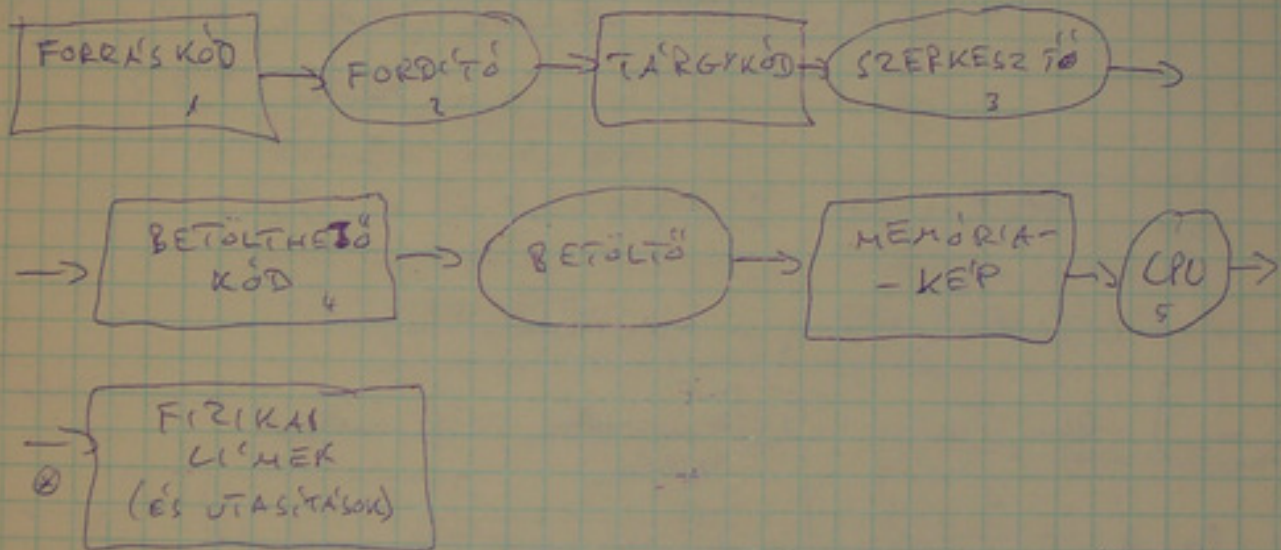
pl.: Hibakeresés azak jó esetben nem lesznek kelutni.

egy programon belüli időbeli dinamikus megváltozás

VIRTUÁLIS TÁRKEZELES

(7+8). óra

12.06



1: mindenholiban változó-nevelés címek helyett

2: más címek vannak, de 0000H-tól kezdődő címmel

3: a szerkesztő összerakja több kódot

/* a fordító minden fordítással pl. C/C++ u. olyan tárgykódot generál */

A realitási övedés a tárolásuk miatt.

6: A kértől eltérő oldal a kért, megmondani melyik a kértés kezdésével. Ezt gyorsítja az ábrák útban. A GT kértésével útban ill. megmondja.

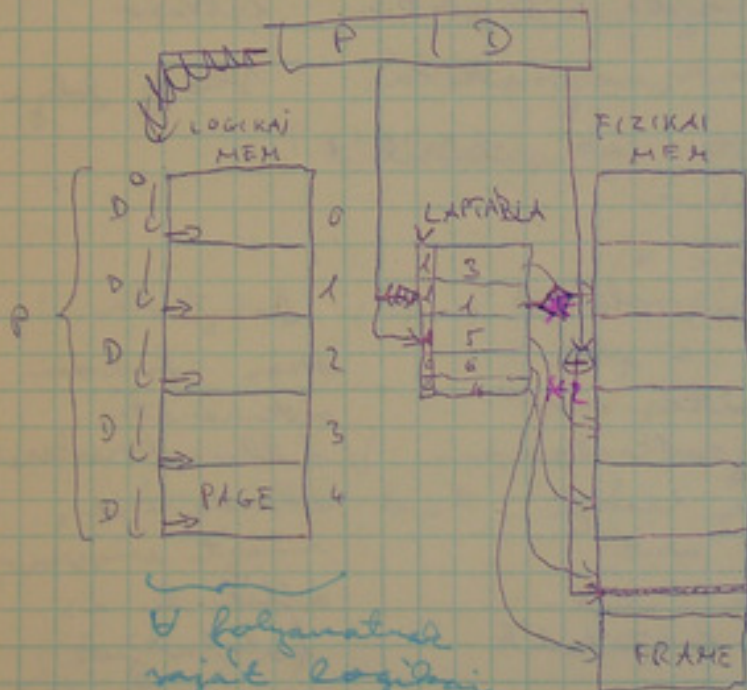
5: CPU: futás közben állandósított RAM, kértés megmondja útban.

Logikailag kértésére fizikailag kértés helyére.

Lapcserés (fix méret)

Memória cserés (változó méret)

1. Lapcserés



P: page number

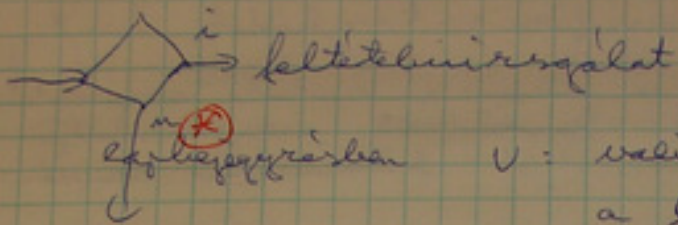
D: displacement

* feltételvizsgálat; túlcserés ellenőrzés

*2 "örvegő" a lap táblában lévő címekkel korlátozva a diszplacment

FIZIKAI cím

A lap tábla rendelkezik örvével a logikai lapok a fizikai tárol. táblával, ami megmondja, melyik logikai lap melyik fizikai frame-ben helyezkedik el.



HIBA
REGSZAKITÁS

V: valid - bit: a lap még a le most futtatott kódhoz tartozik-e

hibanevezés még: pl.: lemez-lekérlelőnél ~~de~~ paritásbit ellenőrzése

hibanevezés: más mint az I/O nevezés, az "hibás" nevezés, 2 utasítás kíséretét fogadja el a CPU; ez jó, mert nem kell statikus menteni

DE hibanevezés: utasítás végrehajtás során fellejő "hibás" nevezés. Hogy folytatam az utasítást?

(1) utasítás rollback: elkezdte a proci az utasítás más végrehajtott részeit, és újra tudja kezdeni azt; utasítás inmetlésevel folytatódik a program

ALTALEX 05

(2) folytatás: ezen finom antinanti állapotváltást elmentése, majd visszaállítás és folytatás az utasítást

RTKd

Multi-programozott rendszer

\forall folyamatosan rajta logikai címre,
kajjai vannak. \Rightarrow folyamathálós?

(1) folyamathálós \Rightarrow lecserelem a
leptáblát

(2) több leptábla van; van egy leptábla
bázis - sorozat, se mutatja meg,
h. egy melyik leptábla érvényes

Fontos a gyorsaság \Rightarrow a leptáblát gyors
memóriában kell tartani

leptábla technika

\forall leptábla a MEM - ben van
és a cache - ben is. (egy sor)

Keverés: két helyen egyidejűleg
indul meg a keverés
/x ha egy tábla nem
fű a CACHE - he x/

\forall memória - művelet 2 x - es ideig
tart: 1 : olvasás a leptáblában
2 : támpelés adal olvasás

Nem kell az egész programot egyre a
HEM - ben tartani.

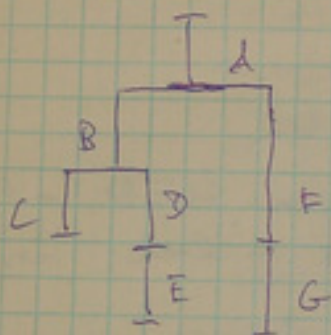


Overlay technika

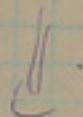
A programok elölr megmondja, egyre
mely programrészeket van szükség \Rightarrow
 \Rightarrow önálló program - modulok



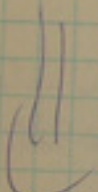
Overlay - fa



} társas : verévlőmodul, "a"
mindig bent van,
innen hívom a
B, C... -t a programból.
Ha nincs bent a
társas, az overlay
manager behozza.



Nem kell az övve modulnévű HEH
tesület, elég a társas + MX modulnévű
hely; persze: idővel több betöltés



1874 MSX \rightarrow virtuális lapkezelés

A teljes címter csak a hirtelen
társa van meg; csak azt a
lapot hozom ki, ha ami kell,
amiről levő utántöltés egy
végrehajtásnak.

A felhasználó / programozó HEH

hellä tiedoksi eräät, joihin kuuluu myös -muut;
 on ajantasainen hardware tärkeitä osia
 intar: IBM OS 360 -näl eläviä

(*)

Elämänsykli: uusia osia, uusia
 van, uusia uusia uusia uusia
 laji; kukaan; uusia uusia
 os uusia, uusia uusia uusia;
 uusia uusia uusia.

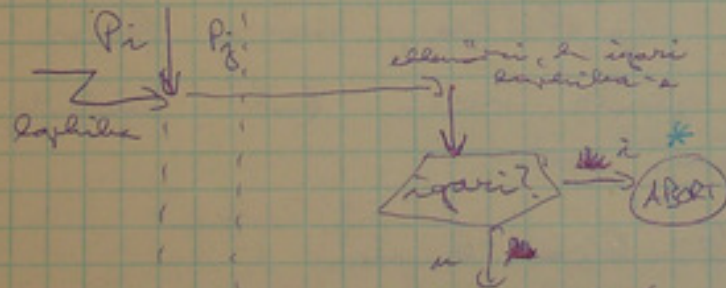
Uusia uusia uusia uusia, uusia uusia
 uusia uusia uusia uusia uusia uusia
 uusia uusia uusia uusia uusia uusia

Uusia uusia uusia uusia, uusia,
 uusia uusia uusia uusia uusia
 uusia uusia uusia uusia uusia uusia
 uusia uusia uusia uusia uusia uusia

- 0 -

CPU

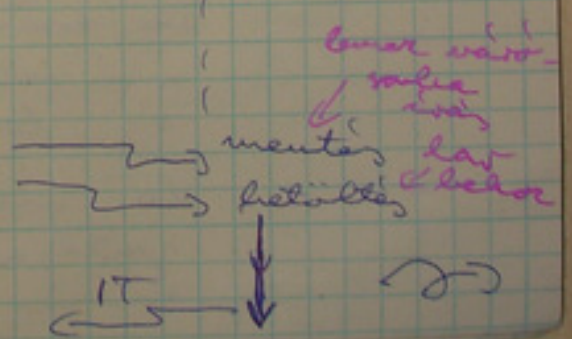
uusia uusia



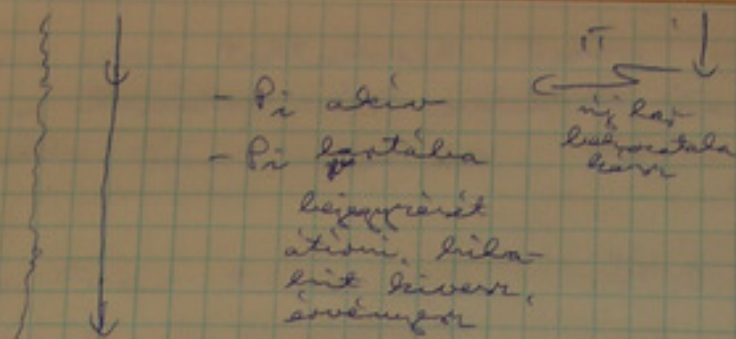
uusia uusia
 uusia uusia

- (1) uusia - uusia
- (2) uusia - uusia
- (3) P_i uusia
- (4) uusia uusia

uusia uusia uusia



* Uusia uusia uusia uusia
 uusia uusia uusia uusia uusia
 uusia uusia uusia uusia uusia
 uusia uusia uusia uusia uusia



Pi futása háss; nagy ríva íl futtatam - e
 az attól függ, h pre-synptise - e az
 intenció

Időrités: Haddig tart egy lapsere?

1 manuskript kb. 10ms

↓
 1 utantás - végelajtas kb 100ms = 10^{-2} s

↓
 Lemereművelés 5 - 10 ms = 10^{-3} s

↓
 Lemereművelés íl sebeség a döntő

Étlagos lapulibrasam?

Ha csak a lapulibria, má a lemer
 vádászva; senki nem fut; minden
 ki lapulibria vár.

Egyszerűen lell kenni!

Lapsere idő alatt, araz lemer-
 művelés alatt ne legyen ~~í~~ sok
 ny lapulibria

10^{+6} memóriaművelés 1 lapulibria
 juthat, nagy egyszerű legyen.

q : lapkita gyakoriasag

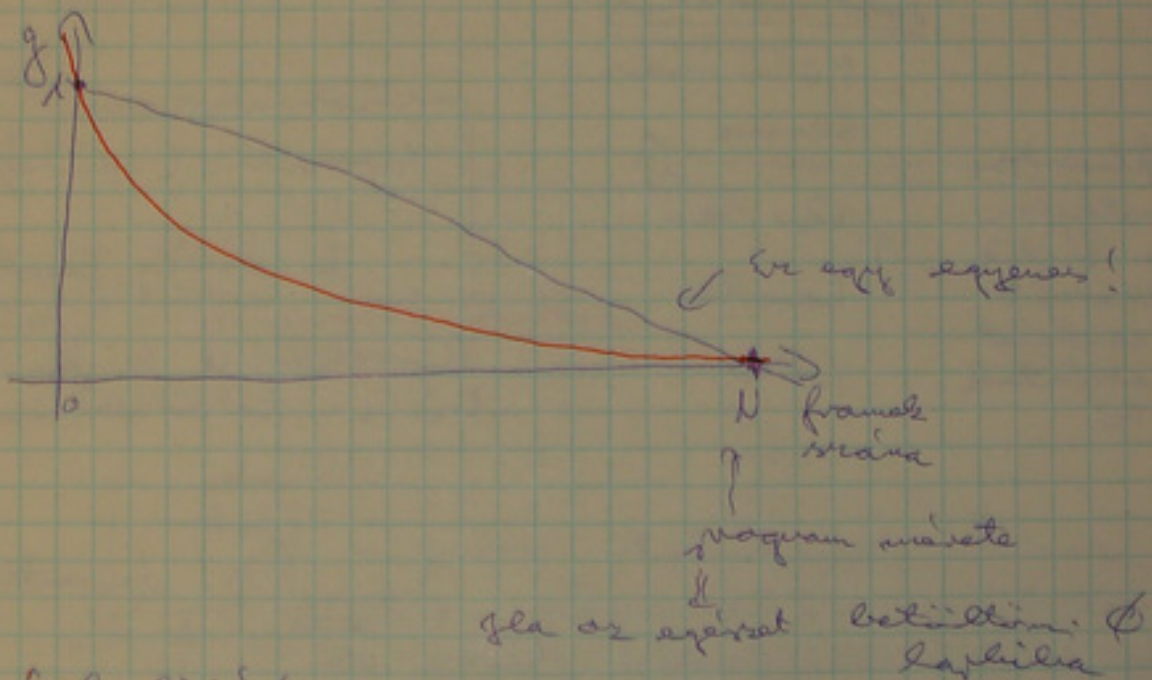
T_{cs} : levezesidő

t_m : max. minelkt idő

$$q < \frac{1}{\frac{T_{cs}}{t_m}} = \frac{t_m}{T_{cs}} \approx 10^{-4} - 10^{-5}$$

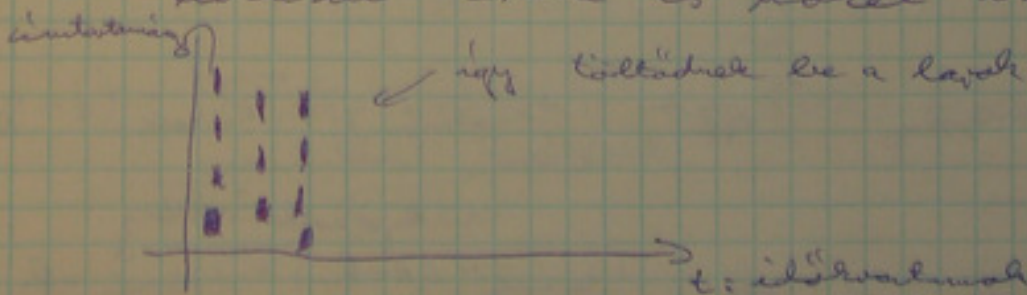
A program rendelkezésére álló mem - méret
N frame - szám & / és a lapkita gyakoriasag
hogya függ össze?

több frame, kevesebb lapkita



Localitás!

A program időben közeli kombinát-
korai tárolás is közel esnek.

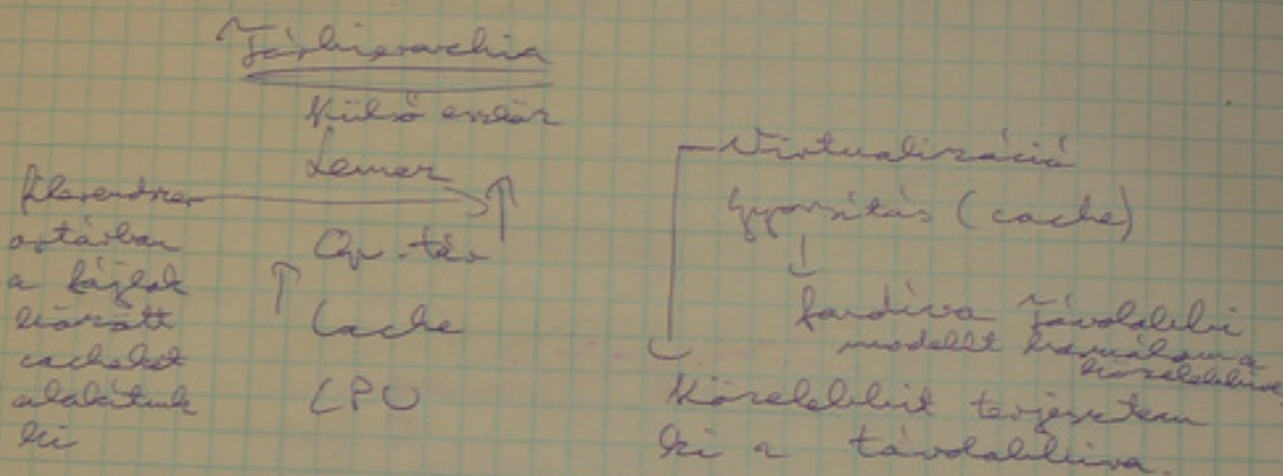


Ha minőségi csapadék után hirtelen
 esőszélvesztés következik, az utóbbi
 két napot négy. A felhős és
 a feltehetően "jövőre"

Adatok: tömörített esőszélvesztés
 ams

rtk
 "hőteljes"

Ha egyszerűen figyelembe vesszük a
 hőteljes értéket a fűtési rendszer



$$\left. \begin{aligned} \text{található mennyiség} &= \frac{q}{\text{össz}} \\ \text{hőteljes mennyiség} &= \frac{\text{hőteljes}}{\text{össz}} \\ \text{effektív elérés ideje} & \end{aligned} \right\} \text{privatizáció}$$

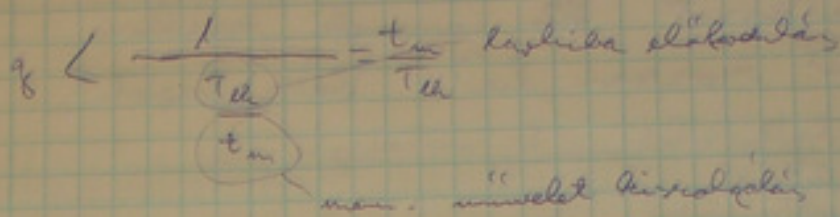
$$q + \text{hőteljes} = \text{össz}$$

$$\text{hőteljes} \rightarrow T_{\text{eff}} = T_{\text{található}} \cdot q + T_{\text{működési}} \cdot (1 - q)$$

ki hirtelen

$$\text{található mennyiség} \rightarrow T_{\text{eff}} = T_{\text{működési}} \cdot (1 - h) + T_{\text{található}} \cdot h$$

privatizáció átlag

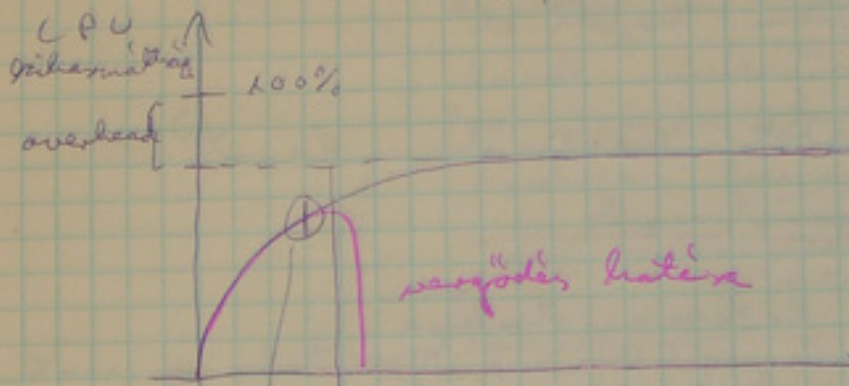


Ha az nem teljessé



Terjedés (Fluviding)

Az az folyamatok beállnak a kover száma és vannak kiszolgálásra. / Kiszolgálás alatt rajta kaphiba gőz és így tovább... */



optimalis

multiprogrammálás
fokozott, rendszeres
folyamataival szembe

konverzió - növekedés \rightarrow csökken az egy
programra jutó aránytérlet

Minél jobban meg kell beosztani a
terjedési határt, DE attól nem
szabad.

Kétféle legyen a két betöltési stratégia?
A hivatkozott állás mindig rendelkezés
közvetlenül legyen az? Kétféle legyen?

↑
lokális

gyakorlati megvalósítás: csak igény szerint
csere van, nem próbálunk előre

↑
latni

demand igény
igény szerint lapok
lokális + lapkészletnél veszünk
figyelme
look ahead
előrelátó lapok
nem tárgyalt el

Kétféle választás szükséges lapot?

Lapcsere algoritmusok

- FIFO lapcsere stratégia
a legkevesebb hivatkozott egy csere

3. LRU: vannak lapok, amiket
bár néha hivatkozunk is, de
folyamatosan hivatkozunk

4. optimalis lenne: megfigyeljük
azt a lapot amire LRU
a legkevesebb hivatkozunk
között; ezt cseréljük
DE nem látunk a jövőbe!

↓
csereink statisztikát

- ha a közelemből van hivathozható
egy lap, választom a jövőben is
hozzát \Rightarrow it nem veszem

legközelebbi hivathozható veszeljél
LRU (Last Recently Used)

Ha hivathozható egy lap, időközleg
vondelim ki hozzá, választásom
azt nézem meg. Mivel a legközelebbi
hivathozható, azt veszem.

Hardwares támogatás kell!

DE nincs a jócska tárhelyben.

- Memóriában tárolással lehet
dolgozni, de az nehézkes.

- Lapszámok szabványos hivathozható
sor elejére hozam; de lassú
hardwares támogatás nélkül.

Közelebbi megoldások
leggyakrabban

Second Chance (SC) = óra algoritmus

listával is
megvalósítható.

DE az nem
hatékony

újabb érvelés
FIFO & LRU alternatívái;

1 bitnyi emlékeztető van
megtárolva, a hivathozható

- e a lapra. preferenciát

V hivathozható 1-ke \Rightarrow listát
a bitet, az előzetesválasztási
algoritmus FIFO szerint választ

magyarul a legközelebbi betöltött

referencia bitjét. Ha az 1, nem vá-
lasztja, a hivathozható lap. \Rightarrow

Udiz lapsed, miig ϕ -t nam talal.
Ist usuli. DE alit meqar, amak
 ϕ -ba allitja a vel. litjet.

tequrunt,
a a kavit
keri kaska
is hivatkon
a va.

\Rightarrow "koleziyo mutato"

Es az altajet, mest FIFO hoz
usak 1 dli jointes dalt, ami mindig
lep 1-et + vel. lit.
egyzam, yvar.

Gyartas: Modontatak-e a lapot?

Ila nem, felalyas, veselo ki-
mentani a memoriaba, mest et
is m. er a jeldany van.

DIRTY - lit

kanyu megalaxitani hardwaron

hikapuram a lapimat, orrel
killetem + dirty - lit fil-
-lapot

lapim + vad's jela

Udoratok valavitanak is fipyalube
f vesem a dirty litet

R : vel. lit

D : dirty lit

RD

00

01

10

11

aljen sorvendlen veselan
a lapokat ; gyartas

torálili (kevéli lémpo) qvntás:

leleka futtassam a kaperese algoritmust,
mikor a CPU vész

|| Grabad pool : kaperamag, vész,
algoritmust futtatam, kaperé néhány
áldozat ^{áldozat} - jelöltet, $\frac{1}{2}$ válszerítam ^{áldozat} $\frac{1}{2}$
a gradad pool - va. kaperé válszerítam
vél innen válszerítam.

↳ láncolt lista jelöltelével

De attól erészet még kaperé válszerítam a
kaperamagot.

Ha válszerítam mégis kaperé válszerítam a
vél. poolon kaperé válszerítam \Rightarrow vél.
kaperé válszerítam \Rightarrow mégsem $\frac{1}{2}$
kaperé válszerítam. listában att kaperé,
de nem válszerítam kaperé válszerítam a lista-
kaperé válszerítam futtatam a vél.
algoritmus.

Klászoknál kaperé

kaperé válszerítam kaperé - kaperé

Ha válszerítam az kaperé válszerítam, kaperé válszerítam
vél a kaperé, kaperé kaperé a kaperé
kaperé válszerítam a kaperé - kaperé.
nem kaperé, kaperé

↳

UNIX

ha a jásziban minos P-list => simuláltság

Simulált referencia-list

1. lapos, lapos, hardwaros
neqreditáral tösléni E nélkül
NDCS vint. táskelés

	HE' hardware évényező list	SE' software évényező list	SR software referencia list
lap listés	1	X	1
GC alap	0	1	0 <small>de a min nem illik</small>
Lapok alap	1	X	1

ha hivalkomak erre a
lapra, leve lapokra
neqreditás DE hat van
a lap a SE' vevint => lapok,
alap. list virtuálitja

Elfoqadhata listakampog-savalsal
neqvalánthata a vint. táskelés.

Lapos algoritmusok működése

Hivalkomai sorral:

Itjen savandhan hivalkomak
a lapokra.

4 frame

his. moral 4 0 1 2 0 3 0 4 2 3 0 3 2 5 6 1 2

50142

Er csak azokat a hivatalosokat tartalmazzuk, amik hivatalos szervek részéről

his. moral

~~4 0 1 2 0 3 0 4 2 3 0 3 2 5 6 1 2~~

4	0	1	2	0	3	0	4	2	3	0	3	2	5	6	1	2
0	4	4	4	0	3		3		3				3	6	6	6
1		0	0	0	0		4		4				4	4	1	1
2			1	1	1		1		0				0	0	0	2
3				2	2		2		2				5	5	5	5

					X		X		X		X	X	X	X	X	X
--	--	--	--	--	---	--	---	--	---	--	---	---	---	---	---	---

↑
franchise
Lapika
o = pontos

0	4	3	3				5	6	1		4
1	0	0	0				0	0	0		0
2	1	1	4				4	4	4		4
3	2	2	2				2	2	2		2
		X	X				X	X	X		X

elavultak, most latjuk
hivatalosok

3 & 4 helyre is
tehetem, egyenlő
közvetlen

0	4	3	3				3	3	1	1	1
1	0	0	0				0	6	6	6	4
2	1	1	4				5	5	5	0	0
3	2	2	2				2	2	2	2	2
		X	X				X	X	X	X	X

0	4+	3+	3	+	+		-3	6+	6	6	-6	4+
1	0+	-0+	-0	+			-0	0	1+	1	-1+	1
2	1+	-1	4+				-4	4	4	2	-2	2+
3	2+	-2	2+				-5+	5	5	5	-0+	0
		X	X				X	X	X	X	X	X

+ : hivatalos
o : FIFO

széleskörű
munka

Fargardallkodar

- globalis

- lokalis

Hány lap kell egy folgyamatnak?

legalább 2 lap tutti kell

egyiken az utastás, másikon az állapotsorok, és a min.

ke utastást végig kell tudni csinálni a vaspados

||

DE lehet indokolttá címsor is
aspados címsorok a címsor
sp. címsor
aspados = 53 lap

||

min. működések lapok száma =>
=> a leggyorsabb utastás is
ke kell tudni futtatni

Hány lap kell a megadott ellenőrzéshez?

Utastás (Működés leírás)
"működés"

↳ ha van lapok száma, melyek

⊕ a program az utastás is
leírás nélkül

↳ ha van lapok száma, melyek

⊗ Irodalom "Költészet" lapok válogatása,
melyekbe a legjelentősebb költészet átka
hivatkozott a program.

Legjelentősebb elnevezés franciaországi fell
a nevezetes elnevezés.

- Gyakori: mások számára is költészet-
ként, mások lapjait is becsel-
tetem a sajátunka.

1. Részlet elnevezés a töltés elöl a lapot

- Lelki: elöl megadom a franciaországi,
amely töltés nem adde melni
Ez nem.

Utolsó: Lelki dinamikus
változása: változtatás
a max franciaországi.

(y+10.) Előadás

12.09.

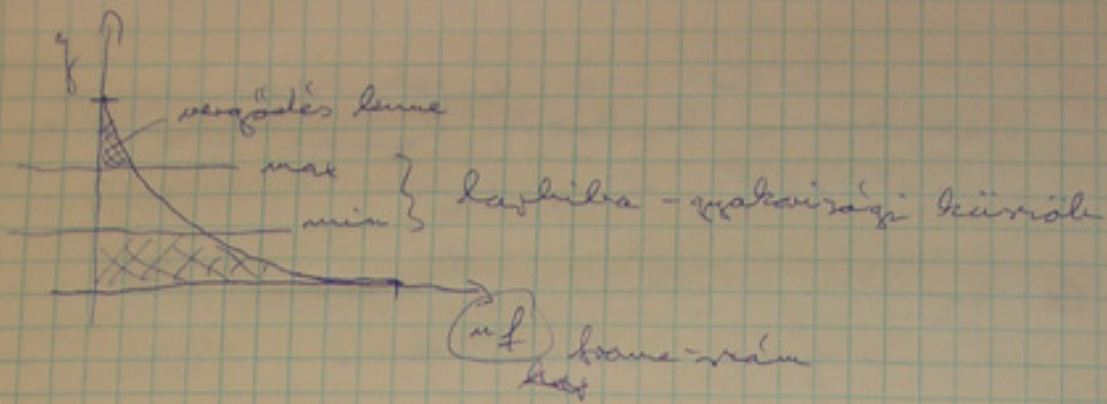
Utolsó Set másképp

al: Folyamat költészet-gyakorlatát egy lealát
al: fell követni

nehézség

DE: költészet-gyakorlat könnye másképp,
errel követni

→



Ha a laphiba - gyakorlati meghaladná a
maxot \Rightarrow vergődés; adjunkt neki frame

— " ————— alá venne a
min - néte \Rightarrow túl kényelmes; töle vesztül
el frame

Távgyedálbeadás: Hányjut frame adjunkt
a folyamatnak a manuális fűtés?

Bélaigy hárszó: magyar; (B.H): vialeralis
kapcsolás egyik fő fejbortója

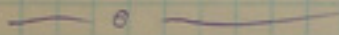
Bélaigy - ananális:

$(n+1)$ frame töle laphiba fordul
elő, mint n frame.

CSAK FIFO kapcsolásnál van ez
és ott sem minden esetben le-
fűtésnél.

1/0 puelet

memória & periféria körüli van
 Hálózatos Hív az 1/0 számon van a leírás,
 addig a CPU - részről nincs használat
 ez a nemtekinthető DE pl.: DMA részről
 használható. Mi van, ha a leírás
 algoritmus leírásai ezt a lapot?
 az csak a CPU - felől & műveleteket
 nézi. Ez GOND. Az 1/0 műve-
 leteket által használhat lapokat ha
 kell kiegészíteni!



virtuális tárolás: felhasználó / programozó
 nem tud róla.

Ez gond is lehet:

pl.: 2 dimenziós tömbök le akarok
 írni

for $i = 1$ to N do

for $j = 1$ to M do

...
 # $T[i,j]$ // deklaráció

pl.: (1) 1 sorozat elem 1 lapot foglal
 új sorozat lapokra \Rightarrow betöltés;
 a leírás címsor 1-1 lépés
 jut 1-1 lapra

fordítási
 műveletek
 ábrázolása

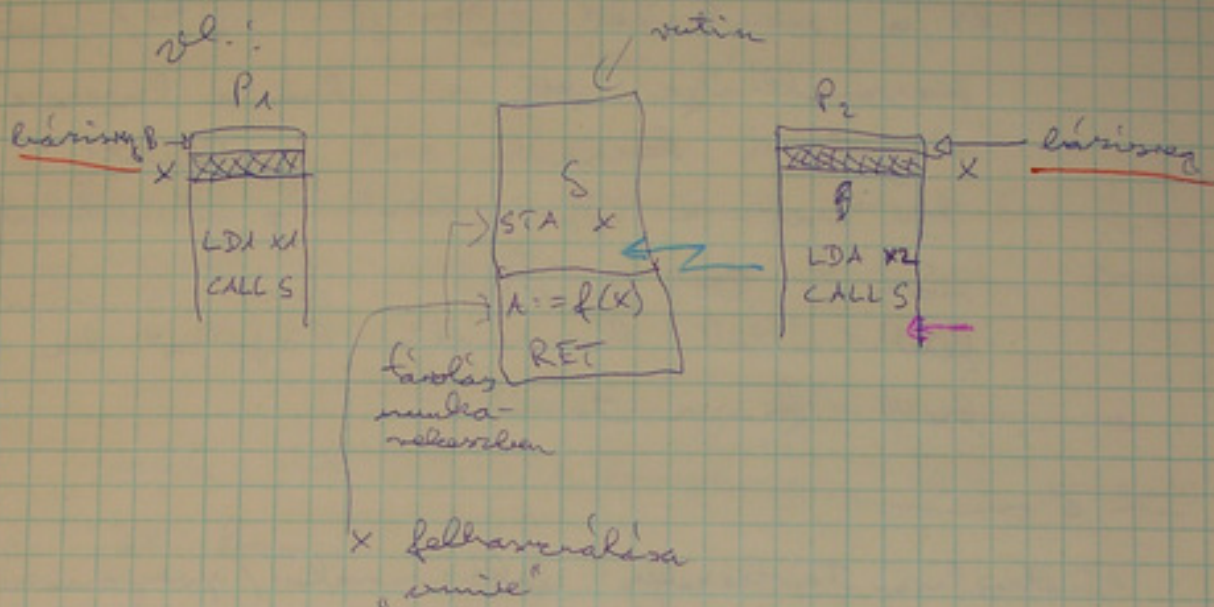


(2) az összerakályozás fordítás;
 1 sorozat 1 lap
 1 leírás címsor minden egyes
 lépés leírásához vezet
 fordítás leírás

feltehetően ha a
 programozó is
 tudja hogy működik a fordítás

Tämäk listios, eljivänsä, amileat table
 program havmal, pl: osuendos, ritiede

Etch njoahivohata ve - entrance kädde.



- P1 meghivata arbritint
- S X värtoröjälual tatalma X1 *
- P2 is meghivöja S-t
- S letävölja X2 tatalmal X-be;
 felililuta X1-at; vissratel;
 P2 - nele OK *
- P1 vissralupja a verälil; S lefut
 is ar X2 alapjan redmitett
 eitelket adja vissra P1 - nele

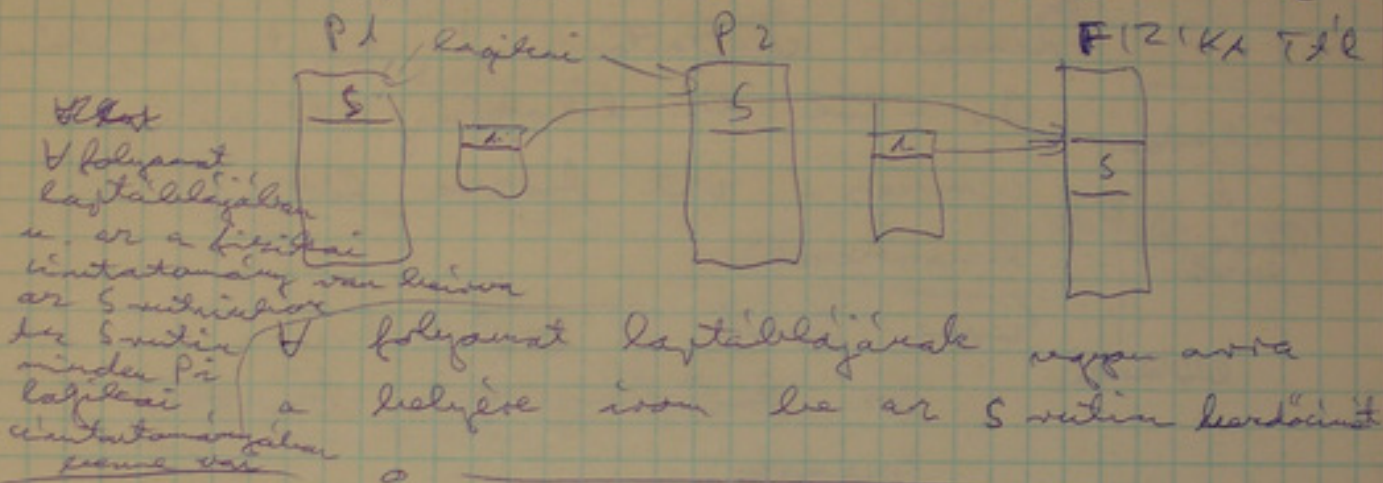
EZ BAZ => Et a lööd NEM
 vissralivohata

hogy ishatunk vissralivohata vutint,
 hogy NWCs -en lelöo vältöröje,
 sem allapottese.

* - Barisrelatív irányú sorok el S-léte az X-at, ~~száraz~~ így mindig a jó folyamati X értéket használom, mert a barispointer átíródik lehet váltással.

- Ez az X a stacken is lehet; a folyamatiak saját STACK megnevezése van.

- A laplábla is a keutextusok tartóik



Lemerkelés; fejléc

↳ alsó réteg; adatokat tudjunk lenni, olvasni

↳ erre épül a fejléc; a lemerkezés fedőrétege

fej: bitosan olvas & ír

1/8 szögben egy fej működik +/

prekvalifikált olvasás & írás egyre,

az 128; 512 stb. byte, attól függ

számszám finamabbi mértékben nem olvasható / írható.

Opisyoner karellat tulle vektort
eggsone; kellest naves kellek

kei
mekk (idindu)
vektor

3 dimensionis ciurea

Ex ar opisyoner karellat

Linearis ciurestamant naves karellat

O. - til N. kellek linearis
ciurea. Art naves letse a
karellat.

Exe ipil a fajvender.

Ex katalogurak, fajvender
naves.

Fajvender
karellat

kellek

Kellek kellek karellat karellat?

idokomponerak:

(1) im karellat, karellat idokomponerak

(2) karellat idokomponerak
ada kellek karellat a karellat

Atlagar adateleseri idokomponerak

11-13 { fajvender
karellat (karellat karellat karellat)
adateleseri

Mivel leírható a blokkok aural
több a hulladékok.

Ha már megvan a blokkok eleje, olvassuk
le mindet töltöt.

DE mennyi adatot kerekít le?

||
Lokális probléma

Teljes egészben sok adatot kell jó
lehozni.

||
Optimalizálás: 4 - 8 - 16 leByte - os
blokkméretek

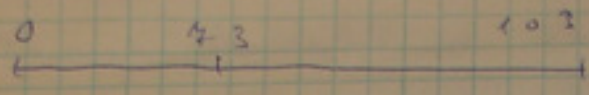
Leműintemérés

hátróssorban blokkok, vonatkozó
leíríteli / leíríteli kéresek.

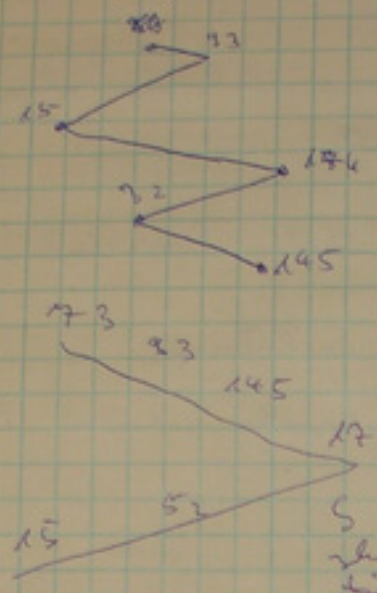
Perifériák vásárlásainál általában
T-IFO intemérés, KIVE'VE a
bent, a kéjesség miatt a
hulladékokat csökkenítették.

kéjesség - minimalizálás: a körüli
(kemény leírati) kéresekkel valóban
a dőrt

||



95
15
174
32
145



- FIFO
vagy
maximálisan
- kivételként
vagy minél
kevesebb
mórozzon

GOND: egy területen
mórozz a
fej||
elcsúsz,

5 57F
rövidebb
time list
nem bair!

Selek algoritmusok: egyirányú halad, elő a névise, majd felvessz a még várakozókat; le-fel járítás és felvessz az utolsó első kivétel

Load algoritmusok: nem megy el a végig ha van más munka kévés; abszolút irányt vált.

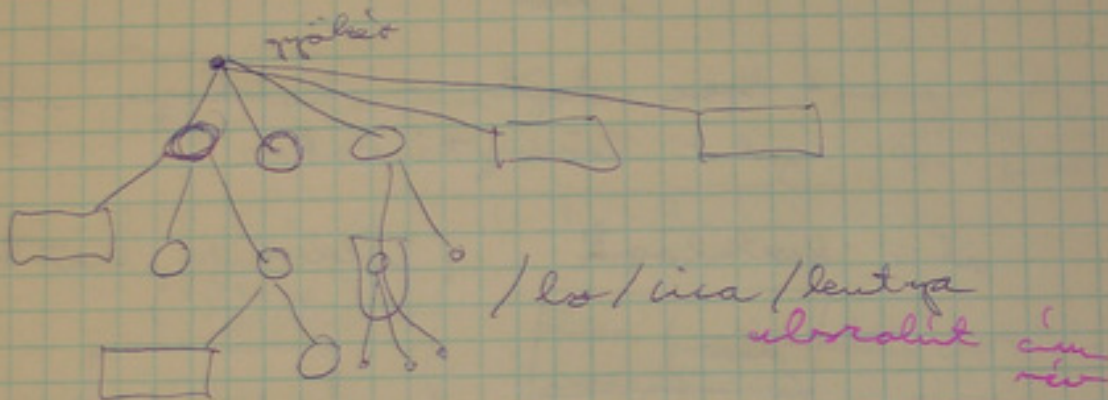
Csele: hovan megijegy, kivételként, majd visszavonul az eljese. Nem ved fel kévésit visszafelé. Csak egyirányú haladás

Load: el. egy, csak visszafordul ha van más munka kévés; irányváltás

Fájlszerkezet

fájl: unikális reprezentáció egybetűzős
adatsorozat

1. Globális csak fájlak voltak, számításkor nem. Egzotikusok lista
2. Később hierarchikus szerkezet; számításkor szerkezet



3. Aktuális számításkor hierarchikus felhalmazokból leötölt megfigyelt fájl, azaz lejáratlanság ide "kezd" a felhalmaz; a számításkor szerkezet felso részét nem feltétlen látja

relatív név

/centra /cont. tet

4. Lehetséges hierarchikus számításkor / fájlak melyek tölele intronálom is elérhetők. szerkezet: hierarchikus - hierarchikus névtelen szerkezet?

Apvondrasat kasinā a failuol
 īmēstēt, mēn aļvājkā arālcāt lēdīlā
 īmēni.

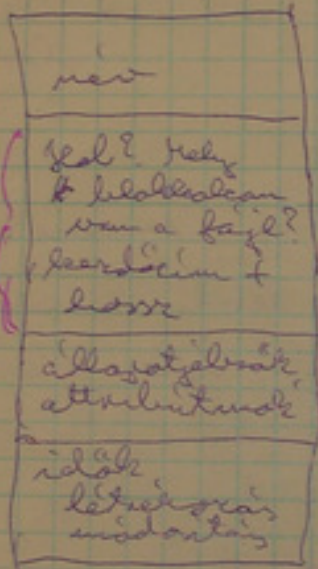
||
 mēn. lētēvārtē

lētēvārtē: vāvān - vāvārtē failuol,
 a sājā fūtētētā ālāvārtē

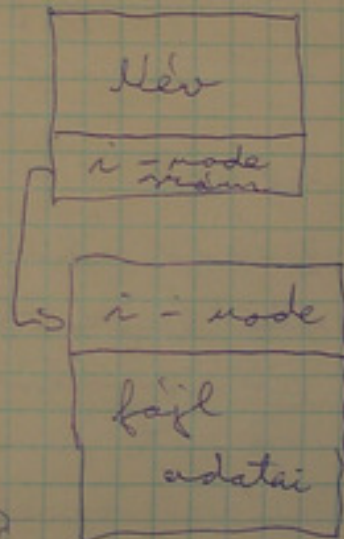
pl.:
 • com
 • exe
 • bat

+ xec fail a kataloģus - fail
 pl.: lētēvārtē

pl.: lētēvārtē



tēlētēvārtē kataloģus
 (UNIX)



Tērtā kā vārtēvārtē
 ā lētēvārtē vārtē
 sājā vārtē lētēvārtē lētē.

ā sājā lētēvārtē tēlētē
 helyvārtē ālāvārtē, a vārtē
 vārtē univārtēvārtē vārtēvārtē
 vārtēvārtē.

Tēlētē kataloģus
 vārtēvārtē a vārtē
 a vārtē i - node vārtē
 lētēvārtēvārtē
 vārtēvārtē, tēlētē
 helyvārtē i lētē
 lētēvārtēvārtē
 vārtē a failuol

Logikai - szint:

Írítók, megírások stb.

Fizikai - szint, szerkeszt

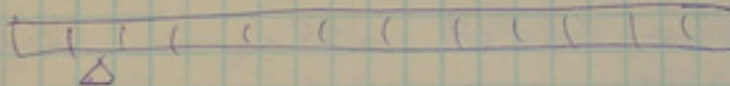
szekvencia - szerkeszt stb.

File - modellek

Fájlok adatainak elérése; ~~file~~ fájlban tárolt adatak elérése.

1) Szekvenciális

magyarszághal (attól a helytől kezdve és a pontok alatt) szekvenciális



read()

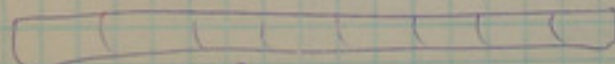
rewind: visszahozza a fájlpointeret aktuális pozícióra és n. olvas, majd lépteti a pointeret

néhány lépés {
logikai: egy szekvenciális lépés
fizikai: egy blokkot lépés
abszolút: léptet lépés

2) Direkt elérés

tudunk címet megadni, az n. helyre

$$\text{read}(n) = \text{seek}(n) + \text{read}()$$



seek(n)

(2)

Allokáció

írd ki a megoldást, mit szokás a feladatoknál?

1, Többszörös

helyezésen van az adat

Metatagok

metatagok



1 helyen csak 1 fajltörés történhet
utolsó blokk elmozdítás helyig van
lehetővé \Rightarrow helyi frekvenciák

fajltörések \Rightarrow helyi frekvenciák

indukciós & direkt elérés is
megelőzhető

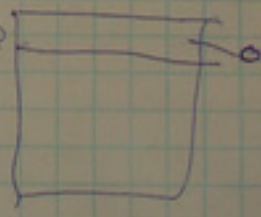
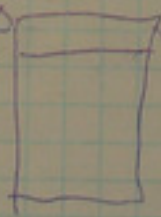
2, Lincolt Allokáció

fajltörés történés adatokat közhívó
tehátam; lincoltá pontosan

Metatagok

metatagok

metatagok



3

Yond: (máslárua blaklaantúell
leerini;

2 lívelit elévist NEH támogetja

(y+11.) Eláda's

12.11.

Lamer frequentia

lívelit

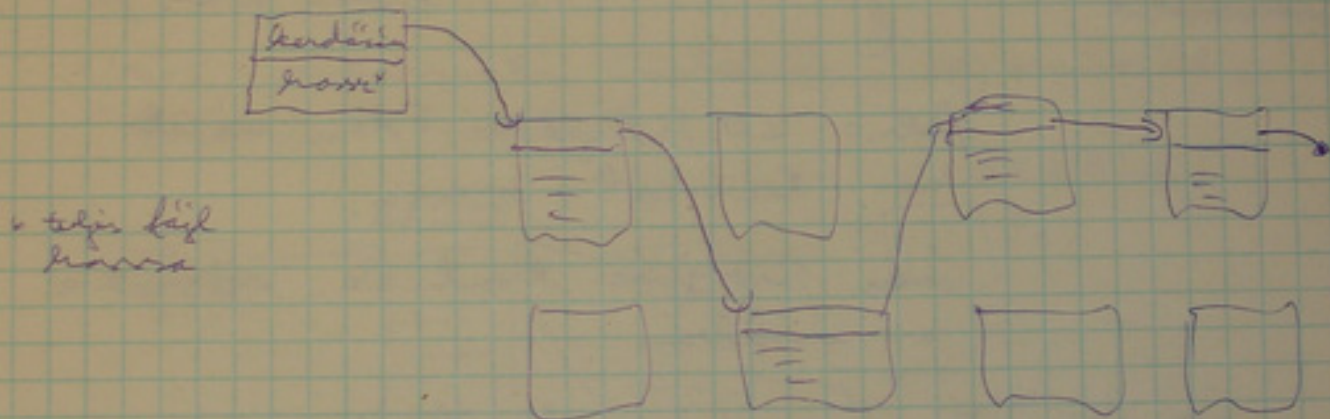
Best fit

stla.

miint manóvialna

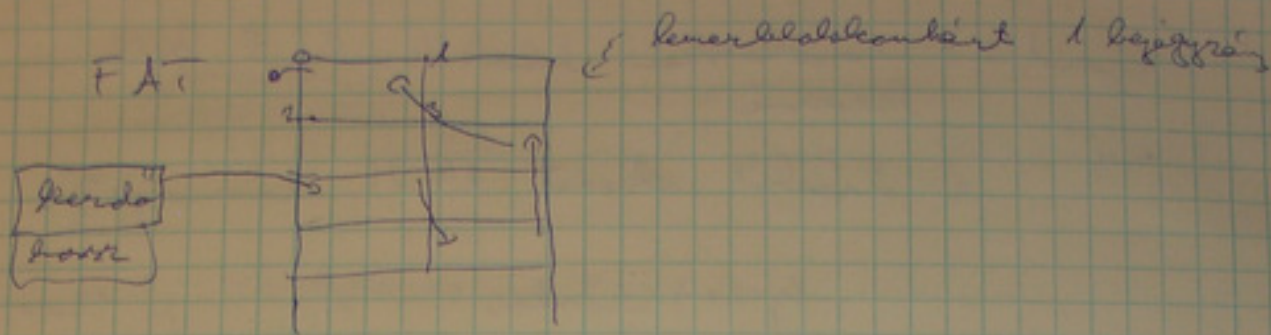
— 0 —
Tolytanos

Láncalt



Yond: lívelit elévist neqia lell
qylogahin a blaklaan

No.: a láncaltó paitenelket NE a
lélla láncalt elemelket tut-
nuk, hanem qyítride lá a
fajl allaladón tállaba.
FTT - etc manóvialnan
tutthata; qyons leeresés.



pl.: $100 \text{ GByte lemez} = 100 \cdot 10^9$

blokkméret $10 \text{ kByte } 10 \cdot 10^3$

blokkok száma $10 \cdot 10^6$ db

lemez a
FAT
2 példányban
másoké
megléni
listaszerű
másként

pl. 20 bit kell egy
blokk címzéséhez

pl.: 24 - bites címzés
32 bites stb.

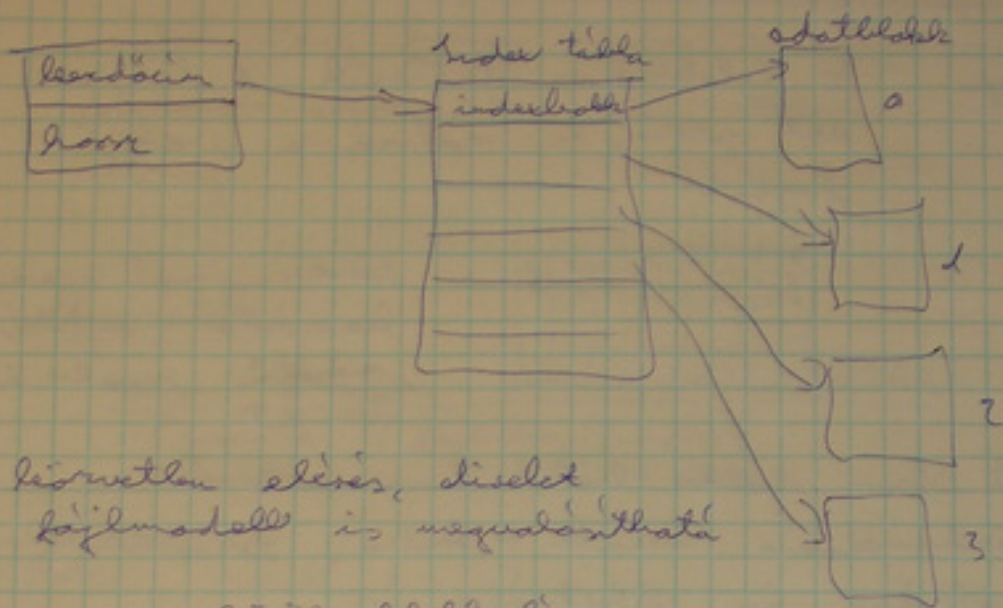
pl.: 40 kByte a FAT mérete
Ez sok, nagy.

Nem tartjuk leant az egészet a
memóriában

gond: Mi van, ha elszakad a lánc?
Össze kell tudni a leszakadt
darabokat

Indexelt állomány:

A fájl blokkjai itt is bárhol
lehetnek, DE kapcsolattáblát
használunk.

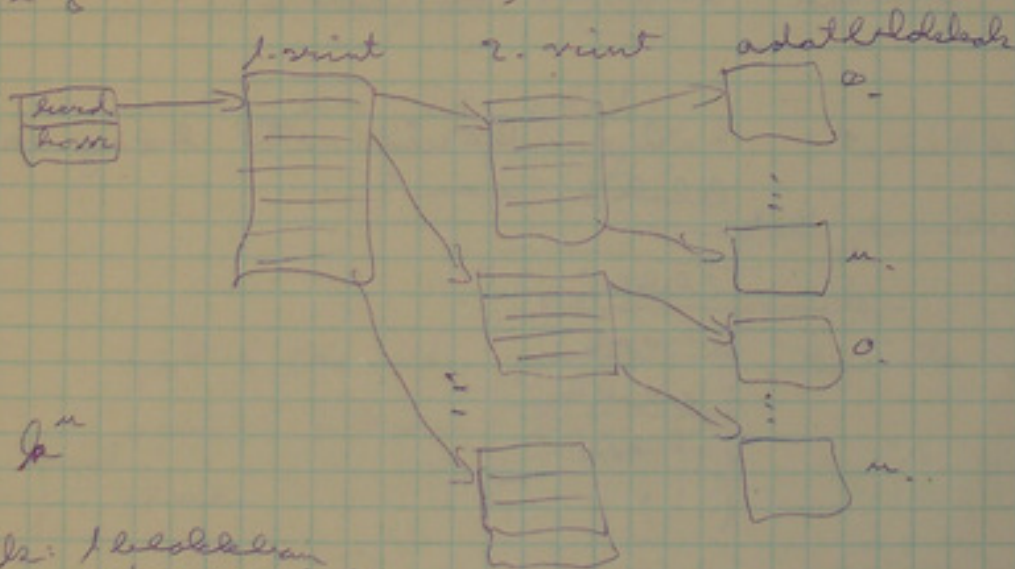


kirjoitteen alies, disect
 kirjasto on isäntä

2560 kiviä koki for
 a tallella

DE indeksit sama lineaarinen nopeus,
 isä koki kiviä kol, lineaaris kirjasto

AVL tallella indeksit, eksponentiaalinen
 kirjasto nopeus



k^n

k: koki kiviä
 kiviä kiviä

n: kiviä kiviä

→

szavak helyét / blokkok elhelyezését

- litterák

V blokkok tartásuk 1 bit, az
azt jelöli, ha szóval-e vagy
szóval

0	0	1	1	0	1	0
1	1	0	1	0	0	0
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

3-dimenziós gráf
"keresési" "megfelelő"
"név" "szavak"
"terület".

szavak több bitet használnak a
redundancia miatt + hibák megelőzése
jelölés is lehetőségek egy

rossz } jelölés
helyett }
szavak }

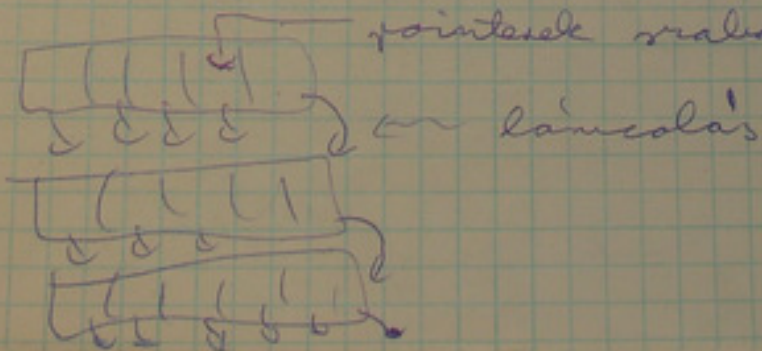
- láncolt listák elhelyezését

szavak blokkok láncolata

- táblázat: milyen kezdőponttól hány szóval
blokkok van?

blokkok
rossz
blokkok
rossz
⋮

- fontosak szavak blokkok



A rendszer a lokalitás miatt mindenképp
 reaktív egy helyen tartani a fájl leíróit,
 folyamatok állapotát; még akkor is ha nem
 mozog.

Műveletek

teljes fájlra értelmezett műveletek

includés a kate- loguson szempont műveletek mint a temp adatok	}	create (fájlnév, (param))
		delete ? leírás
	}	copy
		move

adatokon dolgozó műveletek

read(file-id, adat)

write(file-id, adat)

↑
visszaadott paraméter

↑
beérő paraméter

seek(file-id, pozíció) // filepointer állítása

open(kódnév, file-id)

close

↑
visszaadott paraméter

Év alapján megkérni a fájl a nyitvatartásba,
 és felvétel egy adatcsomagot, a tábla,
 ami alapján már lineárisan elolvasható
 a fájl leírásai. → a read, write,
 már nem leírás a leírás
 a leírás fájl-leírás a memóriában

egy leltő puffertalan tárolás; a read
write - tal ezt a leltőket egyenként
módosíthatom. Ez a leltő puffert
és buffer cache - ként működik =>
=> buffer cache mechanismus.

Védőlelem

- (1) védelem } adatvesztés, adat leavonás
(2) rándékos }

gyarantáltan biztonságos tárolás kell!

/x/ optikail is hasonló problémák x/

lemez - tükrözés : (1) védőlelem
(2) gyarantálás

RAID - tömlő

védőlelem = jogszabály & felhatalmazó arany-
ítás

lényeges, hatékony jól véd
kompromisszumok

leg

UNIX

5.5 FS // System Five Filesystem

Multiprogramozott, több felhasználós rendszer

- Egy alkalmas / egy felhasználó meg-
nyithatja - e n. azt a fájl-típus
vélőnyben. Mit jelent ez?

- n. avva a fájl-típus lehet - e kivételesen
több helyen másként?

különböző felhasználók meg-
nyithatják - e n. azt a fájl-típus?

— 0 —
folyamat - letelepedés => fájl-rendszert
(ös. gyökér)

új folyamat letelepedés: immáron jelölés

1 folyamatok fájl-rendszert is
kommunikálhatnak. * 1 gyökér folya-
mat avval a "munka" által használt
fájlokat

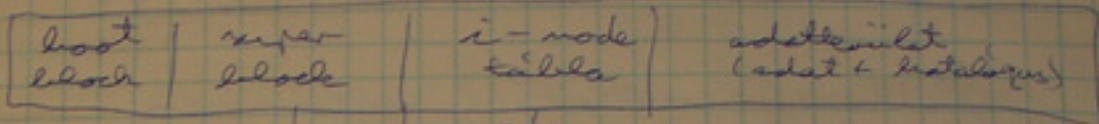
— 0 —

5.5 fs : - lehet megnyitni több fájl-típus

- megnyithatják n. azt a fájl-típus

→

UNIX - keretrendszer



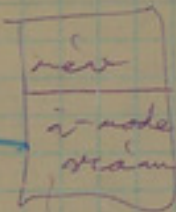
Superblock: Rendelkezővel, az alapján lévő elágazó és a keretrendszer.

Ezzel van egy mátrix a tábla

- rendszerparaméter
- mátrix / matriks - tábla
- szabad hely nyilvántartás
- i-node cache

Katalógus paraméter:

- lehet m. az a fájl tábla helye
- katalógus helye tábla DE file lévő csak egy lehet



az i-node - va mutat;
 az egy lévő, file
 lévő, a fájl minden
 adatát tartalmazza

az alapján
 ezt keretrendszerben az i-node
 táblában.

- fájl - tábla: szabad i-node keresés; i-node cache ezt gyorsítja, ebben szabad i-nodok vannak

- i-node tábla mérete 2^k ; az megadja a tárolandó fájlok max számát

i-node: tartalma:

- tulajdonos (fájl tulajdonosa)
 - csoport (mely csoportba a fájl)
 - típus (read - normal - link; dev / block; pipe; socket)
- ehhez az i-node-hoz van további link

levegőn tárolt adatok

UNIX: 0 / 0 minvelettel
fájl minvelettel
kezelhető
van i-nodja
az 0
erősségnek

soft-link: ez az i-node
egy újabb hivatkozásra
egy más "későbbi" fájlra,
alvétel után van benne
indirekt mutató fájl
név ~~szóval~~

0 erősségűvel

kezelhető vagy lelakos

pl: linkage

pl: link

komunikációs csere

- engedély, jogaitól függően
mind kezelhető mind
minvelettel 3 kategória

keharmälön väpatalak:

tulajdanas

craput

täblilid

müvelakak:

vead

write

execute



H fikshoz

egg 9 kavaktenas

kivä tutonik

	tulaj	craput	täblilid
yl-:	v w x	v w x	v w x
	v w x	v - x	v - -

- idals:

- utdra hovrakas

- - " - modontas

- letshoras

- linkak väma

referenciarvam, az i - modera

untata linkak väma

"Crak egg keharmälötöl
 tövlidit - fikst väp
 tö fikshozilaz is?"

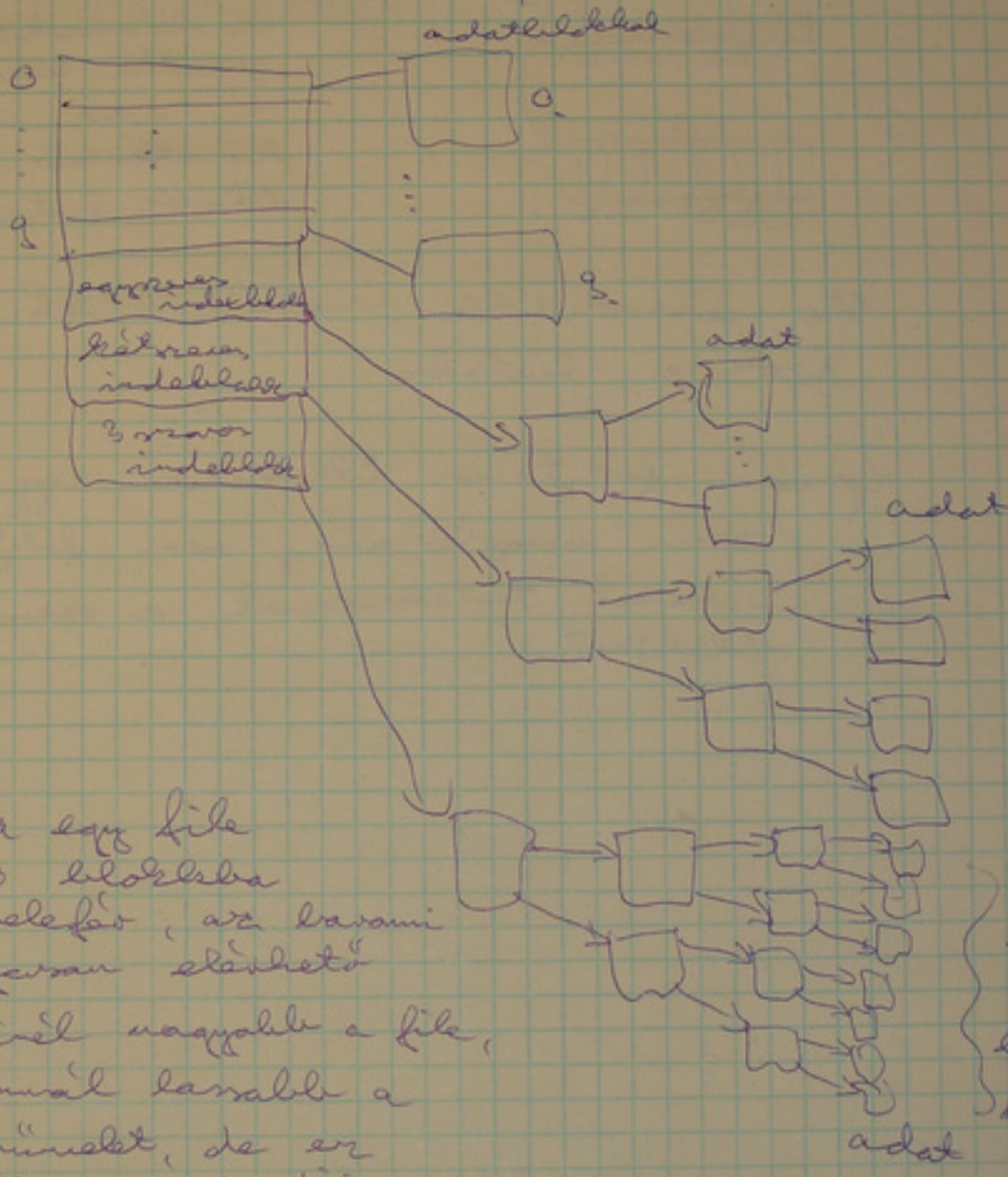
tövlidit a - qvel väkkan a
referenciarvam

a file vad akkan tövlidit, da
a referenciarvam mäa Ø.

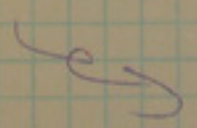
- a file adatarvad elhdjereise

13 db pontot

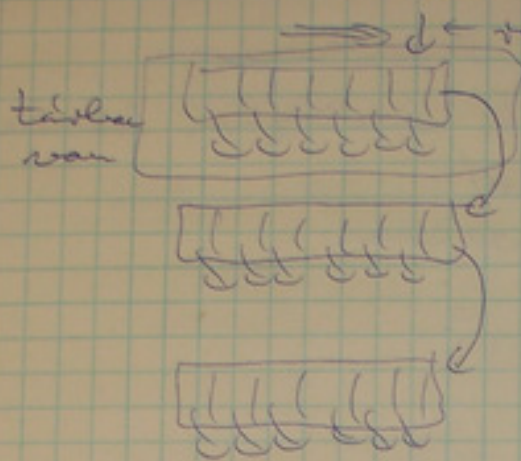
10 db körvetsen hivatkozás
 / * a file első 10 blokkjára körvetsen
 hivatkozással * /



Ha egy file
 10 blokkra
 bontva, azaz három
 egymás alá elhelyezve
 jivel megjelölve a file,
 annál lassabb a
 művelet, de ez
 alacsonyabb szintű
 programok, mint a
 UNIX is.



- szabad hely nyitogatás



pontos: szabad blokkok
nyitogatása

h. ha először
utolsó ámsz,
a láncolás
mentén behozza
a tároló az
kényelmesebb szabad
blokk listát.

Ha felszabadított szabad blokkimeket
isok a listába.

Font: folytonos allokáció így
nem megoldható, nem folytonos
imeket tartalmaznak a szabad
blokkok.