

EXAMEN - 29.01.2024

Nr. 1

1) Dif. var. conditionată vs un semafor?

Semafor (mod de sincroniz)

O construție utiliz. pt. a controla
accesul la o res. prin couterul.

Nu, fol. independent.

Var condiționată

Un mecanism care permi.
unui fol. sa aștepte până
când e cond. devine adeq.

Da, se fol. împreună cu un
mutex.

Caract.

Def.

legat de
un mutex?

3) Prezentă pe 6-biti cu paginare ce emite adrese logice de tip p1 d
 a) Care e nr. total de pag? Care este dimensiunea unei pagini care este
 dim. totală a meiu. virtuale?

$$p = \text{pag.}, d = \text{dim.}$$

$$\text{nr. pag.} = 2^2 = 4 \text{ bytes}$$

$$\text{dim.} = 2^4 = 16 \text{ bytes}$$

$$\text{dim. tot. meiu. virt.} = \text{nr. pag.} \cdot \text{dim.} = 4 \cdot 16 = 64 \text{ bytes}$$

$$8 \text{ bits} = 1 \text{ byte}$$

$$4) \text{int - 2 bytes} \Rightarrow 2 \cdot 10 = 20 \text{ bytes}$$

$$V[10]$$

De căte frame-wri e meiu? ~~1 pag~~ 1 pag. = dim. unei pag.
 $20 : 16 = 1, \dots \Rightarrow$ deci 2 frame-wri
 (rotunjirea superior)

c.a.c. r.m. set.

Apare fragm.?

Dacă apare virgulă (nu se sup. exact) \Rightarrow fragm. internă

5) Tabla de pag; unde $V[5]$? cum arată adresa?

logică pt. acces

4 pag.-luate de la(a)

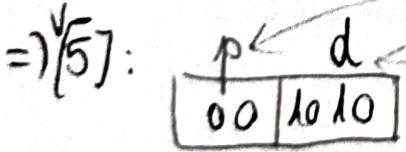
$V[6]$	$V[8]$	$V[9]$		
$\dots V[7]$	111100			

16 bytes

fragm.

un int-2bytes

scup de
set 0



$\Rightarrow V[5]$ se află pe pag. 0 (prima)

$V[0] \quad V[1] \quad V[2] \quad V[3] \quad V[4] \quad V[5]$

2 2 2 2 2

10

offset

c) Câte copii putem face max. în meiu?

Nu pot începe copia din fragm. \Rightarrow ne interesează nr. de pag.

$V[6]$	$V[8]$	\dots	$V[9]$	$V[8]$	\dots
$\dots V[7]$	$V[9]$	\dots	$V[7]$	$V[9]$	\dots

$\Rightarrow 4 : 2 = 2$ copii max.

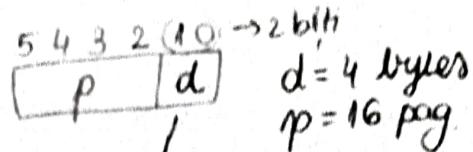
(la fel și dacă $5 : 2 = 2$)

Cum ar merge pag. pt. a stoca nu multe? Cate copii ar incapa at. ? Eude
↓
to. modif. btr. pt. pag. si diam.
- ar ajuta v[er]g?
Adresa log pt. acce?

EFICIENȚĂ: dimensiunea h. să fie div. cu (20) și putere a lui 2
div. rest.

\downarrow alegeri 2 sau 4

Aleg. 4. Refacere desenul:



20:4 frame-wi

$$16:5 = 3 \text{ copii}$$

11 pag.

mär.
pag

* frame \leftrightarrow fog menu pt.
 (menu) (menu. virt.)
 pt.

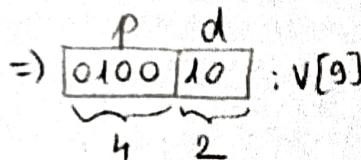
W01,W11 = -2 [4 5 6 +18 9]

4 buyer

11
pag. 4

me int doar pe prima pag. (iii)

↳ are 1 int in fatā $\Rightarrow 1 \cdot 2 = 2$ Offset



Nu există fragme.⁽⁴⁾

4) Sec. de ph în lista de ast.

Process	Start	CPU
P ₀	1	4
P ₁	0	3
P ₂	3	5
P ₃	2	8
P ₄	3	1
P ₅	2	4
P ₆	4	5

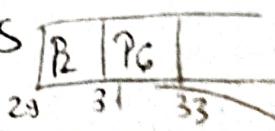
$\text{CPU} = \frac{\text{tf. total}}{\text{tf. exec.}}$

Alg.
Schedul.

Round Robin

—S7F

FCFS



P ₁	P ₀	P ₅	P ₃	P ₄	P ₂	P ₆	P ₀	P ₈	P ₅
0	3 4	6	9	12	13	16	19	20	21

$$P_0(4, g_0) \leftarrow P_5(4, g_0) \times P_3(3, g_0) \leftarrow P_4(4, g_0)$$

diagr. Gant, $g_0 = 3$

$$P_G(2, g_0, 2) \rightarrow P_k(2, g_0, 2) \xrightarrow{P_k} (S_1, 20) \xrightarrow{\frac{15}{2}} \text{punkt}$$

4) SJF? prezyd.

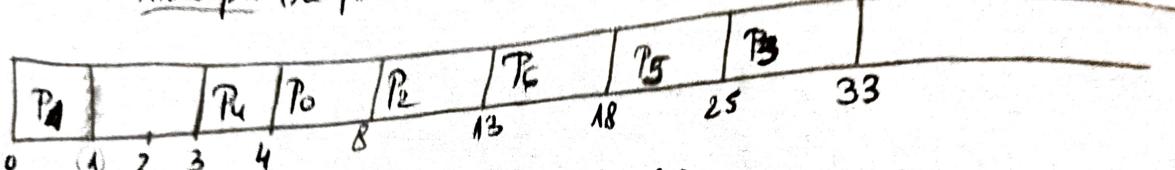
moc-pi: (le pun in ord: 3-6-8-9 →)

Burst time ~ CPU

(cât tp. sta pe procesor)

3)

R
C



compr. P₁ mai are 2D
P₀ mai are 4D ⇒ coadă: P₀(4)

mocu. 2: P₁ 0 D

P₃: 4D

P₅: 8D

mocu. 3: P₂(5)

P₄(1)

mocu. 4: P₆(5)

=> P₀(4), P₃(8), P₅(7) (irrel. ord.)

SJF = Shortest Job First

c) tp. de aștept. mediu?

✓ 13-3

$$(a) AWT = \frac{[0+13]+0+(10+13)+(7+12)+9+15+27]}{7} : 7 = 15,428$$

$$(b) AWT = \frac{[3+0+5+22+0+16+9]}{7} : 7 = 7,857$$

5) sist. de fis. cu strategie de alocare bazată pe indexare pe 2 nivele unde un bloc are 128 bytes impl. pe un CPU pe 16biti unde un byte = 8 biti.

a) Căți indexi per 1 nivel? În total? Care e dim. max. a unui fișier?

EXAMEN - 02.02.2024

1) Care e dif. intre un proces si un fiz. de executie? + Ex scrie pt. a arata cum ar putea fiol. 2 pt. cu miiile de capac. of. de 2 fiz. de exec. si vice-versa.

PROCES

- o instanță a unei programe în execuție
- proprietăți specifice memoriei, variabile, resurse
- comunicare între ele și prin pipe-wri, mem. partaj., mesaje

FIZ. DE EXECUȚIE

- o unitate mai mică de execuție în cadrul unei procese
- fizice din același proces împart memoria și resursele acelui și
- comunicarea între fiz. e mai rapidă

2) Coadă de aștept. pag.: 4, 5, 4, 1, 3, 2, 6, 1, 3, 2, 6, 5, 2. În coadă fiec. nr. reprez. identif. unei pag. ce va adăuga în meniu. pt.?

(a) LRN, care e nr. min. de frame-wri necesar pt. a nu prod. page fault?

(b) Fie n acest nr., cum arată alg. pt. n-2 frame-wri?

a) 4
5
1

ii)	4	5	4	1	3	2	6	1	3	2	6	5	2
	4	4	4	4	4	6	6	6	6	6	6	6	6
	1	5	5	5	5	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1	1	5	5
	1	1	1	1	3	3	3	3	3	3	3	3	3

n=6
4 frames
il scot pe cel din mai trecut

compl. la final

EXAMEN

* Proc. (CPU) pe 8 biti. = 1B

4) FAT \rightarrow 3 fisiere: foo, bar, baz

- numele fisierului: max. 8 caract. pt. nume \times 3 pt. extensie \Rightarrow 3 blc. de start
- un bloc are 64 bytes, 1B = 8 biti

a) FAT tîmînd pe un bloc, cîte fisieri putem stoca max?
Care e dim. max a unui fisier? Dar pe 2 blocuri? Arătați tabelele.

$$8 + 3 + 1 = 12 \text{ bytes}$$

\hookrightarrow (intre extensie și fisier propriu.)

$$\text{CPU} = 1 \text{ byte}$$

$$1 \text{ bloc} = 64 \text{ bytes}$$

\Rightarrow pt. fiecare fisier avem nevoie de

$$12 + 1 = 13$$

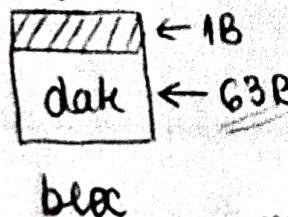
dim.
nume dim.
pointer

1 fisier. NU e pe 2 blc.

$$\Rightarrow 64 : 13 = 4, \dots \Rightarrow \text{max. } 4 \text{ fisiere/bloc}$$

Pe 2 blocuri (tîmînd cont de fragmenții) avem 8 fisiere max. (4+4).

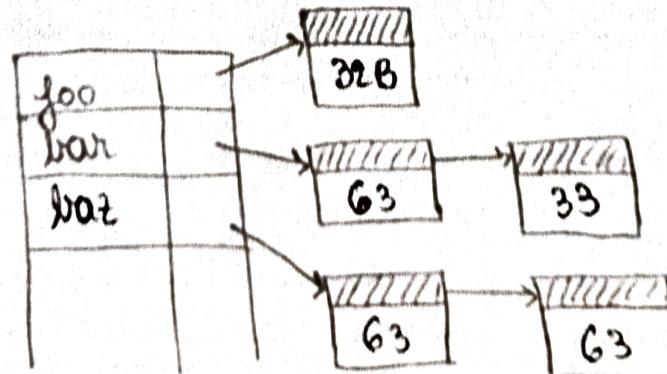
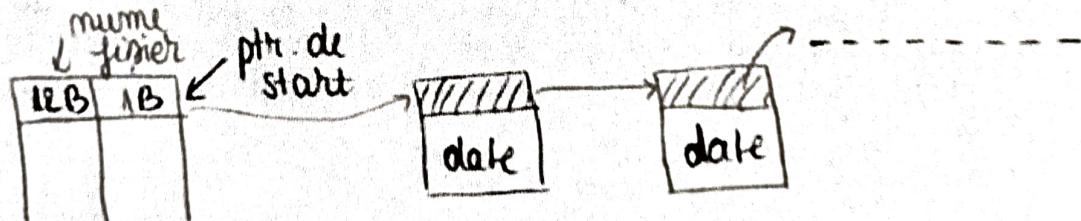
Dim. max. a unui fisier e dată de cîte tipuri dif. de pointeri
putem genera?



\Rightarrow putem genera 2^8 pt. unici

$$\Rightarrow \text{dim. max.} = 2^8 \cdot 63$$

* tabela FAT:



i) foo - 32B

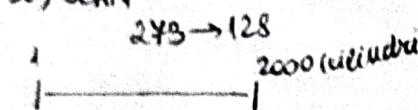
bar - 96B

baz - 140B

5) disk cu 2000 cilindri
capă de cerere I/O în apt. 512, 1024, 96, 1880, 256, 1500, 365, 2
frec. intrare reprez. un cilindru
capul de citire disk: 128 și a fost mutat la 273

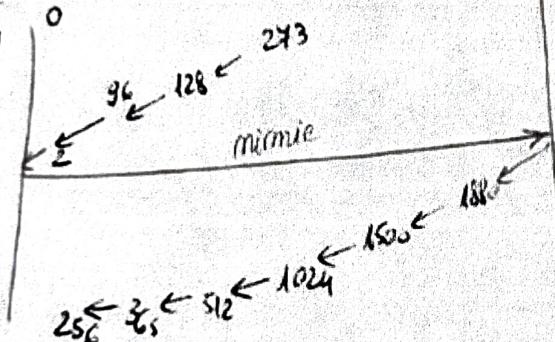
3) a) incoporand de la poz. curentă → ord. și dist. totală (alg. FCFS)
 $273 \rightarrow 128 \rightarrow 512 \rightarrow 1024 \rightarrow 96 \rightarrow 1880 \rightarrow 256 \rightarrow 1500 \rightarrow 365 \rightarrow 2$
 $\text{dist.} = (512 - 128) + (1024 - 512) + (1024 - 96) + (1880 - 96) + (1880 - 256) +$
 $+ (1500 - 256) + (1500 - 365) + (365 - 2)$

b) SCAN



$$d = (128 - 96) + (96 - 2) + (2 - 1) + \\ (256 - 1) + (365 - 256) + (512 - 365) + \\ + (1024 - 512) + (1500 - 1024) + \\ + (1880 - 1500)$$

c) C-SCAN



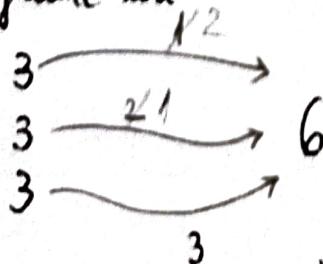
$$d = (128 - 96) + (96 - 2) + (2 - 0) + \\ (1999 - 1880) + (1880 - 1500) + \text{ord.} \\ + (1500 - 1024) + (1024 - 512) + \\ + (512 - 365) + (365 - 256)$$

3) instr. reg., memop, memop (ex: add r8, [0xdead], [0xbeef]).

a) Nr. min. de frame-uri?

PCB ← Regs
(Pr. Control Block) INSTR reg (IR) → inst. la care se afer codul
1 frame pt. adresa de mem. min. (sau 2)
L ⇒ 1 frame/proces

b) 3 pt. cu căte 3 frame-uri
au doar 6 frame-uri



$$\Rightarrow 1 + 1 (+1)^* = 2 \text{ sau } \underline{\underline{3}}$$

thrashing → un pt. nu poate să
inc. tot pe mem.

⇒ { page fault nou y,
sau scoate ceva (1/0)

⇒ N tot scot între ele

2) 4,5,4,1,3,0,6,1,3,2,6,5,2

a) Nr. min. de frame-uri (OPT) sără page fault?

7, deoarece avem 7 pagini dif. Dacă am avea mai puține, la un nou dat ar trebui să facem o modif. \Rightarrow dici page fault.

b) $m=7 \Rightarrow 5$ frames

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

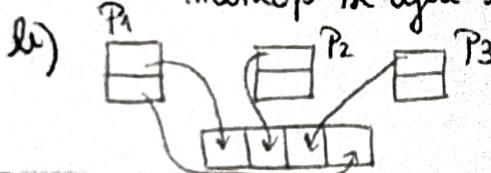
! te uită împreună că apare cel mai departe de unde ești!

3) instrucție reg, memop, imm (ex: add r8, [0xb01d], 0xface).

a) Nr. min. de frame-uri?

imm este codat în instrucție } = 1 frame
instrucție se află în blocul text al procesorului } = 2 fr.
(txt din PCB)

reg și legătura pe procesor = 0 frame
memop se află în memorie = 1 frame



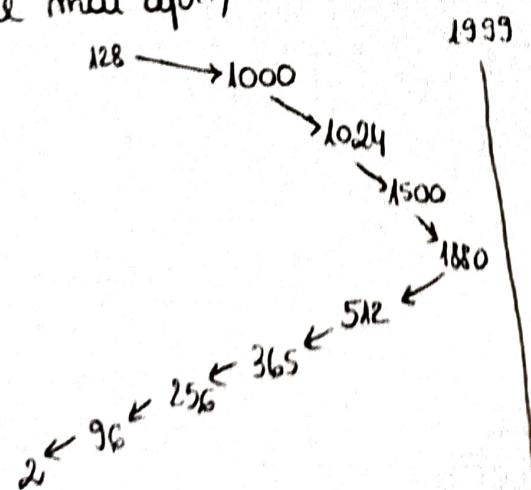
Când al doilea p. din P2 inc. să intre în sezon pe altul \Rightarrow se prod. page fault-uri deoarece astfel se pierde f. mult tp. pe op. I/O \Rightarrow thrashing.

5. disk cu 2000 cilindri

cereți: 512, 1024, 96, 1880, 256, 1500, 365, 2

poz. act.: 1000 \leftarrow 128

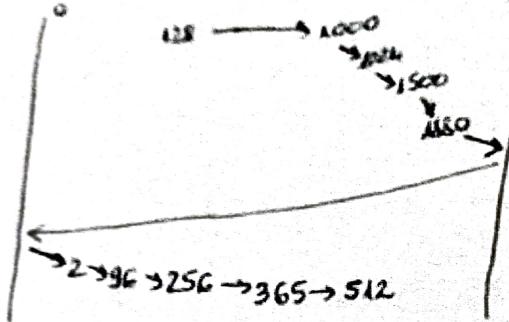
a) SSTF (cel mai apropiat)



\Rightarrow Ord. din desen.

$$\begin{aligned}
 d = & (1024 - 1000) + (1500 - 1024) + \\
 & (1880 - 1500) + (1880 - 512) + \\
 & (512 - 365) + (365 - 256) + \\
 & (256 - 96) + (96 - 2)
 \end{aligned}$$

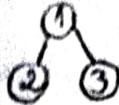
(a) C-scan



$$\begin{aligned}
 d &= (1024 - 1000) + (1500 - 1024) + \\
 &\quad + (1180 - 1500) + (1999 - 1180) + \\
 &\quad + 1999 + 2 + (96 - 2) + (256 - 96) + \\
 &\quad + (365 - 256) + (512 - 365) \\
 \Rightarrow d &= 24 + 446 + 380 + 119 + 2001 + 999 \\
 &\quad + 160 + 109 + 447 = 500 + 499 + \\
 &\quad + 2095 + 416 = 999 + 2511 = 3510
 \end{aligned}$$

=> Ordinaria: 1024, 1500, 1880, 2, 96, 256, 365, 512

2) FIFO



1) procese:
threads:

2) Procesor pe 8-bitii: $\underline{\quad 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0}$



a) nr. total de pag. $\Rightarrow (p =) 2^5 = 32$

dim. unei pag. $\Rightarrow (d =) 2^3 = 8$ bytes

dim. totală a mem. virt. = nr. pag. dim. $= 2^8 = 256$ bytes

b) int - 4 bytes

$v[9]$ de int $\Rightarrow 9 \cdot 4 = 36$ bytes

$1 \text{ byte} = \text{dim. unei pag.}$
(addr. fiz.)

(addr. log.)

De câte frame-uri e nr.?

$36 : 8 = 4, \dots \Rightarrow 5$ frame-uri

Apare fragm.? Da, este intermită (nu se împ. exact).

Tabel de pag.:

8B	{	<table border="1"> <tr><td>$v[0]$</td><td>$v[1]$</td></tr> <tr><td>$v[2]$</td><td>$v[3]$</td></tr> <tr><td>$v[4]$</td><td>$v[5]$</td></tr> <tr><td>$v[6]$</td><td>$v[7]$</td></tr> <tr><td>$v[8]$</td><td></td></tr> </table>	$v[0]$	$v[1]$	$v[2]$	$v[3]$	$v[4]$	$v[5]$	$v[6]$	$v[7]$	$v[8]$	
$v[0]$	$v[1]$											
$v[2]$	$v[3]$											
$v[4]$	$v[5]$											
$v[6]$	$v[7]$											
$v[8]$												

Adr. log. pt. $v[7]$: pe pag. 3

Are rămânețe în val. \Rightarrow 4 ~~bytes~~

(doar în pag. respectiv)

$\Rightarrow v[7]: - \underline{p} \ \underline{d} \ | \underline{100}$

c) Câte copii ale lui v pot avea time max. în meciu?

Nu pot copia dim. fragm.
incepe

32 pag.

1 vect. \Rightarrow 5 pag. } $\Rightarrow 32 : 5 = 6, \dots \Rightarrow$ maximum 6 copii

Cum modif paginarea pt. a fi cu multe? (scopul este ca dim. unei pag. să fie div. lui 36 și putere a lui 2)

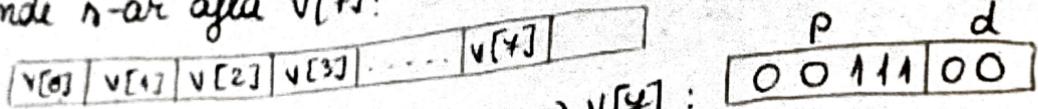
Modif bitul pt. $P \oplus d$

$$\Rightarrow \begin{array}{ccccccccc} & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \text{dim.} & | & | & | & | & | & | & | & | \\ \text{vect.} & P & - & d & , & 1 & 0 & & \end{array} \Rightarrow d = 4 \text{ bytes}$$

$$p = 2^6 = 64 \text{ pag.}$$

Aveam: $36 : 4 = 9$ frame-uri $\Rightarrow 64 \text{ pag.} : 9 \text{ frames} = 7 \text{ copii}$
 dim. dim. unei pag.

Unde se ar afla $v[?]$?



\Rightarrow pag. + de la 0 (offset) $\Rightarrow v[4] :$ (adr. log.)

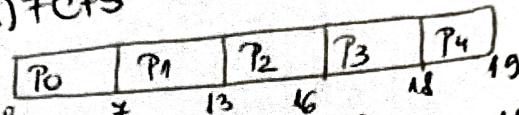
p	d
0	0 1 1 1 0 0

EXAMEN - Série 24 - 2020

Nr. 2

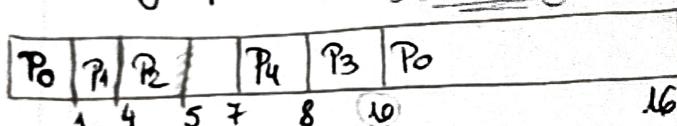
	Start	CPU
P ₀	0	4
P ₁	1	6 3
P ₂	4	3
P ₃	5	2
P ₄	7	1

a) FCFS



$$AWT = \left[0 + 6 + \frac{(13-4)}{9} + 11 + 11 \right] : 5 = 37 : 5$$

b) SJF preempt. \rightarrow le restău cu modif.



(P₀, 6), (P₃, 2)

$$AWT = ((0 + (10 - 1)) + 0 + 0 + 3 + 0) : 5$$

c) RR cu modif. \rightarrow q = 2 (de dubl. când rese)

SJF non-preemptive (sortate nu scot)

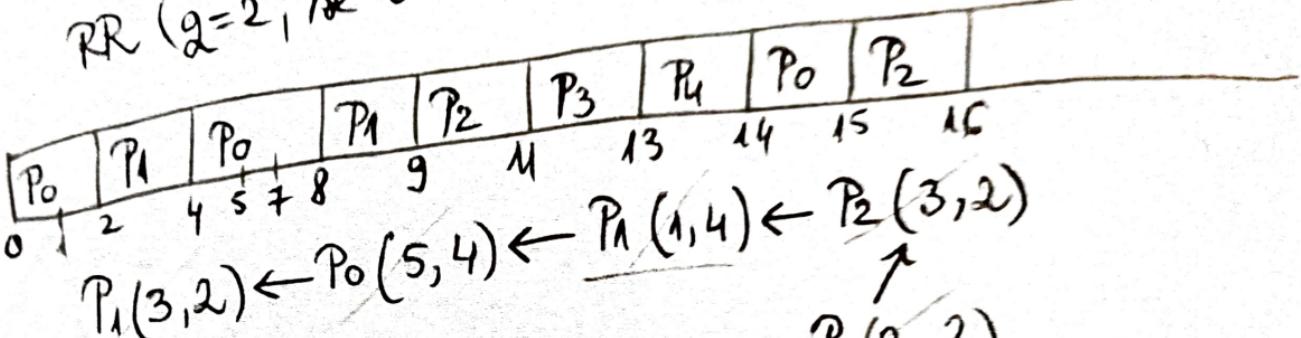


- in ord. Burst Time
- din alia care au intrat
- deja pâmă la mow. +

scoti de pe P₀ \Rightarrow ult. în a

Process	Start	CPU
P ₀	0	4
P ₁	1	3
P ₂	4	3
P ₃	5	2
P ₄	4	1

RR ($q=2$, ∞ duleară)



initial & final

$$AWT = \left[((0 + (4-2) + (4-8)) + \right. \\ \left. + ((2-1) + (8-4)) + \right. \\ \left. + ((9-4) + (15-14)) + \right. \\ \left. + ((4-5) + (13-4)) \right] : 5$$

$$\xrightarrow{P_2(1,4)} \xrightarrow{P_0(1,8)}$$

$$\xrightarrow{P_4(1,2)}$$

$$\xrightarrow{P_3(2,2)}$$

$$\xrightarrow{P_2(3,2)}$$

$$\xrightarrow{P_1(1,4)}$$

$$\xrightarrow{P_0(5,4)}$$

$$\xrightarrow{P_0(1,2)}$$