TD SE

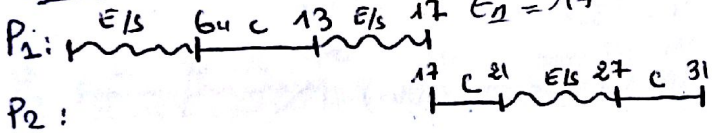
1:

- P_1 { 6 µ E/S ; 7 µ calcul ; 4 µ E/S }
- P_2 { 4 µ calcul ; 6 µ E/S ; 4 µ calcul }

1 - Mono programmation: ordre P1, P2

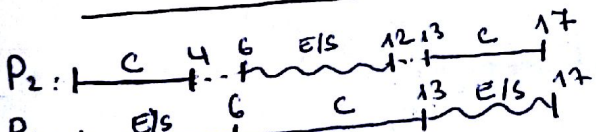
$t_1 = 17$
$t_2 = 31$
$\boxed{tm = 24}$

vraie..



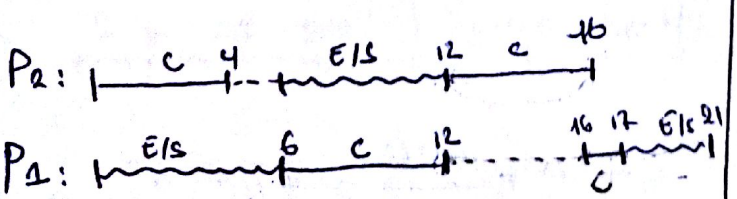
2 - Multiprogrammation: ($P_2 > P_1$)

1 - Sans IT programme:



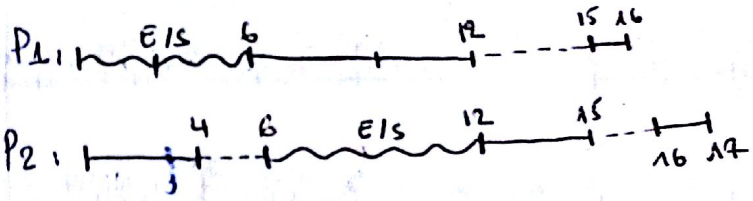
$t_1 = 17$
$t_2 = 17$
$\boxed{tm = 17}$

2 - Avec IT programme: $P_2 > P_1$



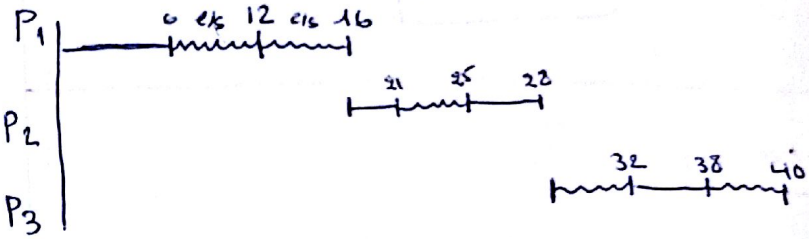
$t_2 = 16$
$t_1 = 21$
$tm = 18,5$

3 - temps partagé: (3/4)



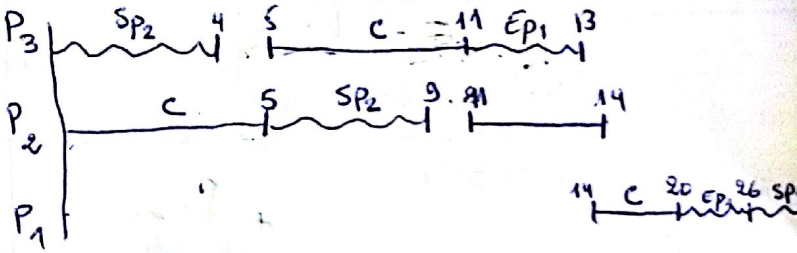
EXO n°2:

1) Monoprogrammation (ordre P_1 P_2 P_3)



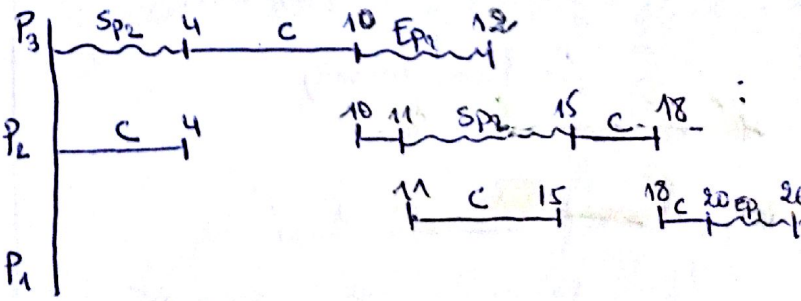
$\boxed{tm = 28}$ $((16 + 28 + 40)/3)$

2) multiprogrammation sans IT de programme : ($P_3 > P_2 > P_1$)



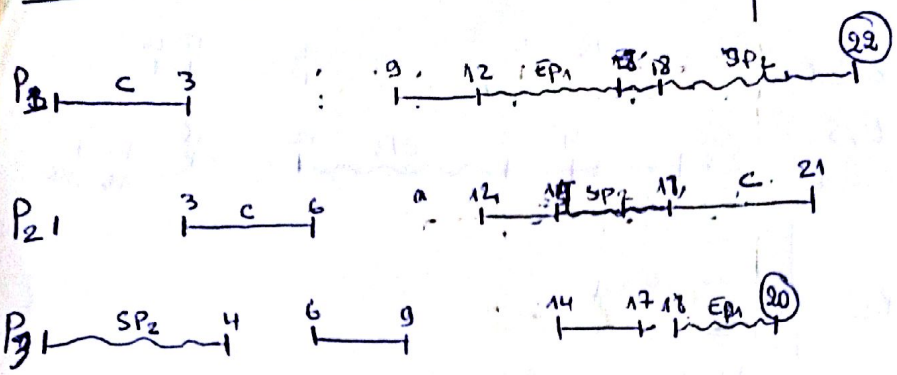
$\boxed{Tm = 19}$ $((14 + 13 + 30)/3)$

3) multiprogrammation avec IT: ($P_3 > P_2 > P_1$)



$\boxed{Tm = 20}$

*4) Temps partagé (3u)



$tm = 21$

$n = i - 1$.

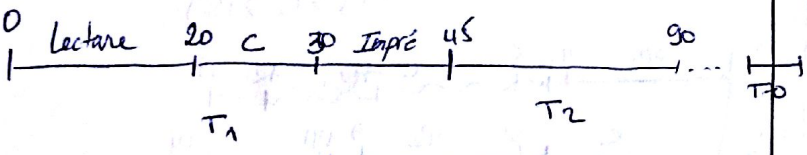
$t_i = t_{i-n} + n(20)$

$t(i-i+1)$

Exo n°3:

1 = l'UC gère les E/S: (pas de simultanéité entre E/S et le calcul)

(monoprogrammation)

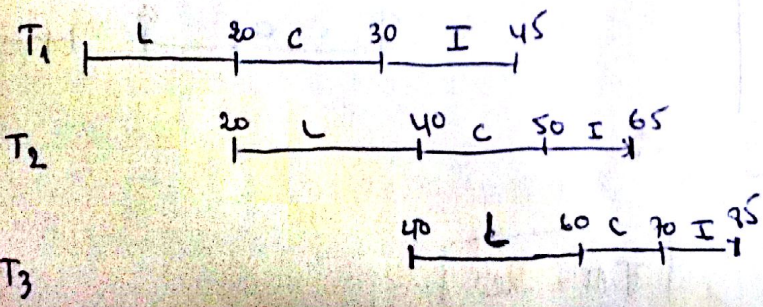


$t = 45 \times 70 = 3150s$.

Taux d'utilisation de l'uc = $\frac{\text{calcul nbr d'instructions} \quad 19 \times 70}{3150} = 22,22\%$

2 - les périphérique sont autonome:

(multiprogrammation)



donc:

$t_i = t_{i-1} + 20 \quad \forall i > 1$.

$t_i = t_{i-2} + 2x(20)$

⋮

$t_i = t_{i-n} + n(20)$.

En posant $n = i - 1$:

$t_i = t_1 + (i-1)\,20$.

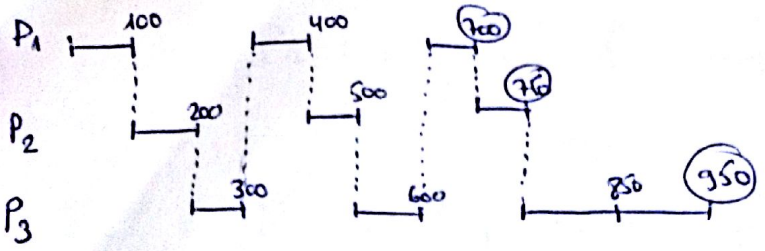
$t_{70} = t_1 + 69(20)$

$= 45 + 69 \times 20 =$

$t_{70} = 1425$

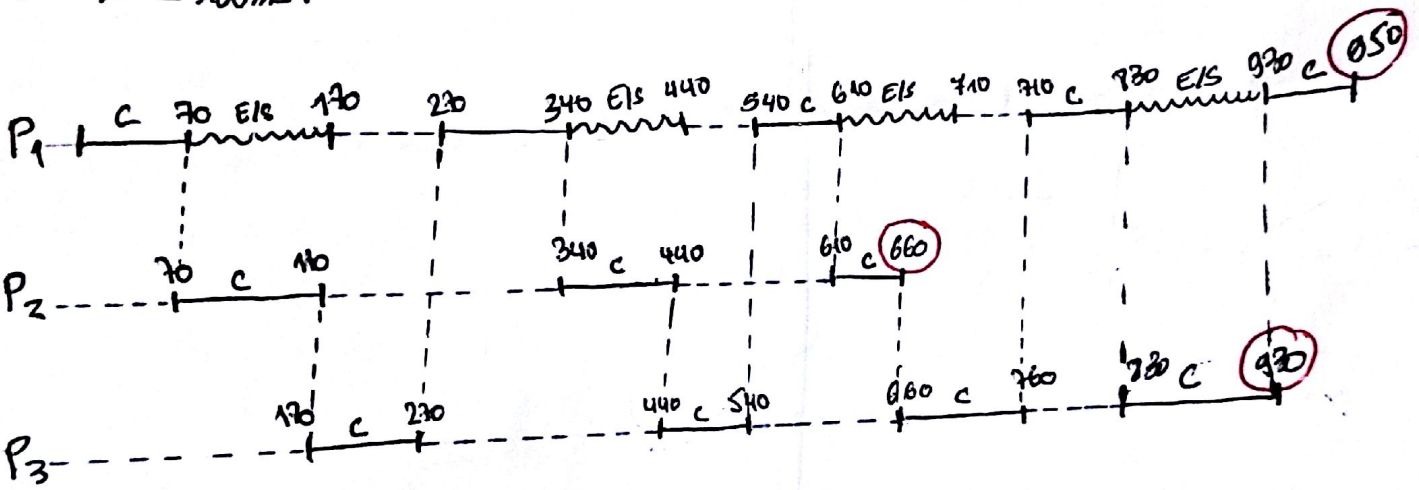
$taux = \frac{7000}{1425} \simeq 49,12\%$

$t_{P_2} = 250s$, $t_{P_3} = 400s$.

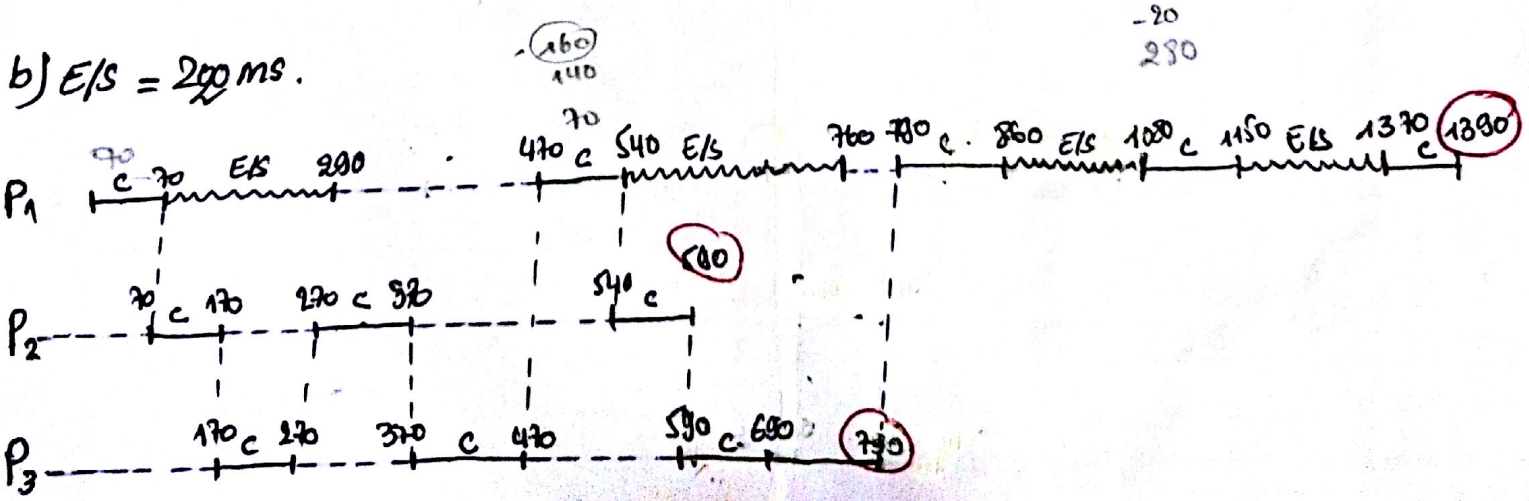


II) P_1 se bloque en E/S toutes les 70ms:
(càd: a chaque 70 ms de calcul il s'arrête pour faire des E/S)

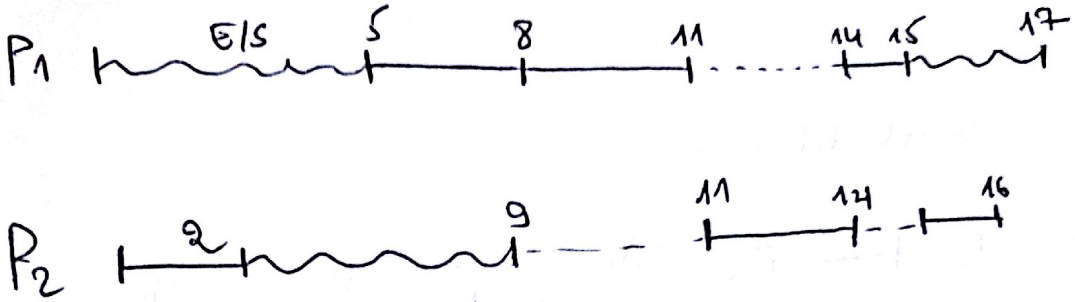
a) E/S = 100ms.



b) E/S = 200 ms.



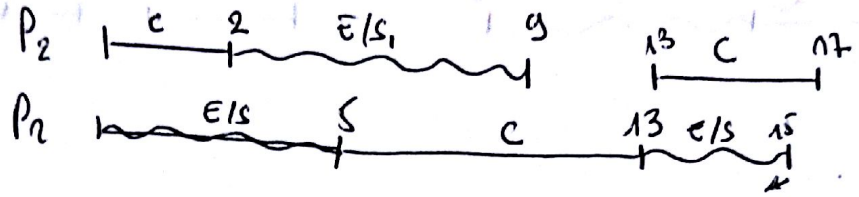
temp partagé



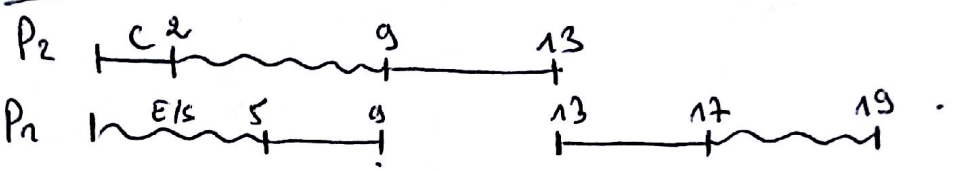
P_1 { 5u E/S1 ; 3u C ; 2u E/S1 }

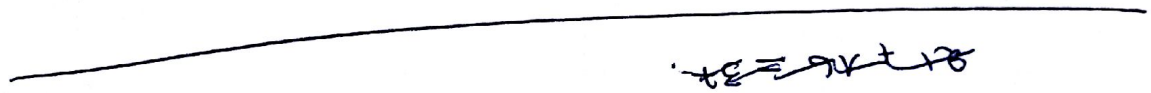
P_2 { 2u C ; 7u E/S2 ; 4u C. }

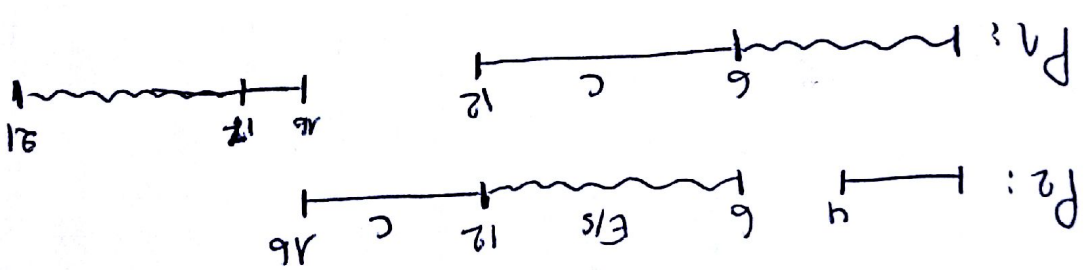
Sans IT.

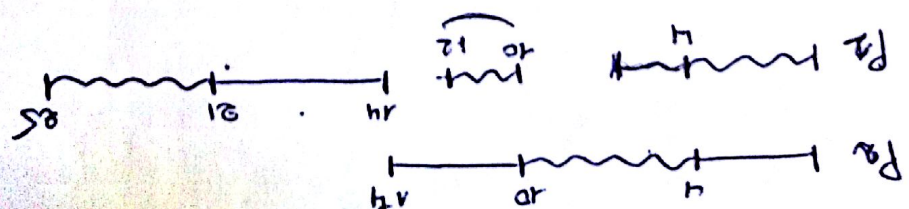


Avec IT









$E_{c_2} = 44$

$E_{c_1} = 85$

TD N°2

	CHA	Zone	$0000
	ADD	deux	$0003
	COMP	Max	$0006
	BEG	fin	$0009
	ADD	trois	$000C
	BRI	Suite	$000F
fin	RNG	result	$0012
	F. :		$0015
Zone	DEC	17	$0016
deux	DEC	2	$0018
trois	DEC	3	$001A
result	RES	2	$001C
max	DEC	26	$001E

pour fin : on ajoute que 1, car il n'y a pas d'adresse, il y a que le code opéran donc il est sur 1 bit.

Pour DEC : DEC n'est pas une instruction mais une pseudo-instruction donc, on ne le considère pas comme étant "code operation", il n'y a que l'adresse donc DEC xxx est sur 2 bits

RES : même chose avec DEC

table des Symboles :

Nom	Adresse	type
Debut	0000	Label (étiquette)
Suite	0006	label
fin	0012	label
Zone	0016	mot
deux	0018	mot
trois	001A = 26	mot
result	001C	mot
max	001E	mot

pour debut

CHA Zone
100 | 00 16
Code operation — adresse (d'après table de symb

ADD deux	02 | 00 18
COMP max	06 | 00 1E
BEG Fin	04 | 00 12
ADD trois	02 | 00 1A
BRI Suite	05 | 00 06
RNG Result	01 | 00 1C
FIN	03
Zone DEC 17	/ | 00 | 11
deux DEC 2	/ | 00 | 02
trois DEC 3	/ | 00 | 03
result RES 2	/ | / | /
max DEC 26	/ | 00 | 1A

Rem :

La dernier instruction commence à

00 1E et se termine à 00 1F

plus le 0000, donc en tout on a

$(0020)_{hex}$ = 32octets

On a besoin de 32octets pour

tous le programme.

Un assembleur à une passe :

on va établir deux tables,

une pour les instructions assemblées

et une autre pour les instructions

non - assemblées

Inst assemblées.

Inst machine	Adresse		
5 0006	000F	JBRI Suit.	
99	0015	Fin	
0011	0016	Zone	DEC 17
0002	0018	DEC 2	Deux
0003	001A	DEC 3	Trois
	001C	Rés 2	Result
007A	001E	DEC 26	Max

Inst non-assemblé

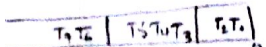
Inst machine	Adresse	
00, 0016	0000	← CAR
02, 0002	0003	← BUF
06, 001E	0006	← MAX
04, 0012	0009	← BEGIN
02, 0003	000C	← ADD
01, 001C	0012	← RNG

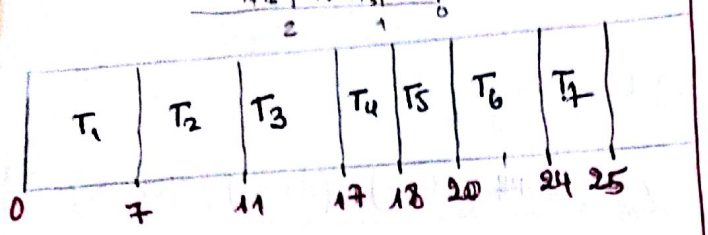
Scanned by CamScanner

2ème Partie (TDN 4)

Exo n°1 :

1) FIFO :



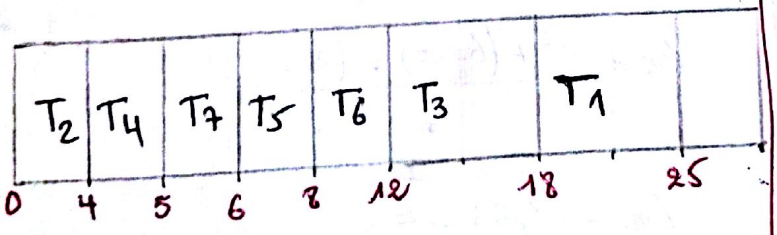


Temps moyen :

$$tm = \frac{7 + 11 + (17-1) + (18-1) + (20-1) + (24-2) + (25-2)}{7}$$

$tm = 16,42$

2) PCTE (ou SJF)

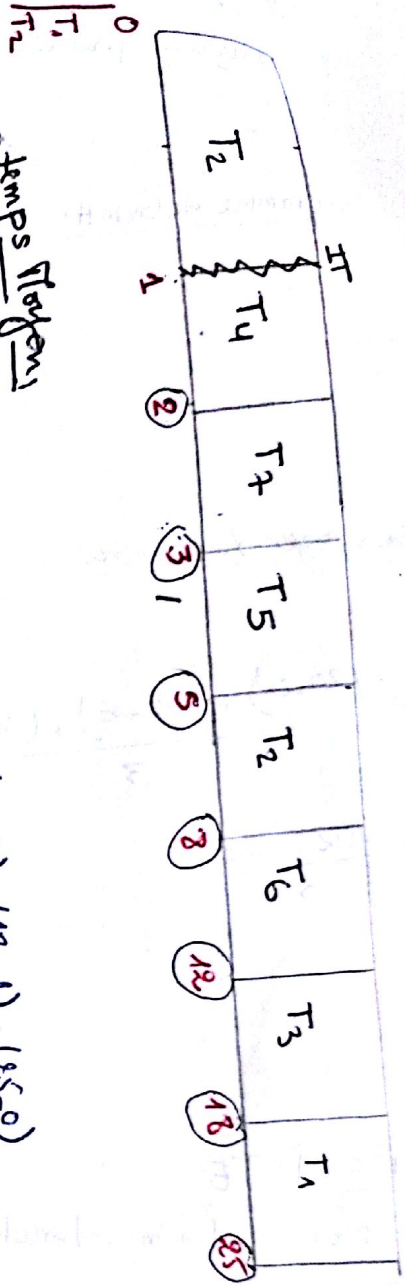


Temps moyen

$$tm = \frac{4 + (5-1) + (6-2) + (8-2) + (12-2) + (18-1) + (25-0)}{7}$$

$tm = 10,14$

3) PCTER (Plus Court temps Restant)

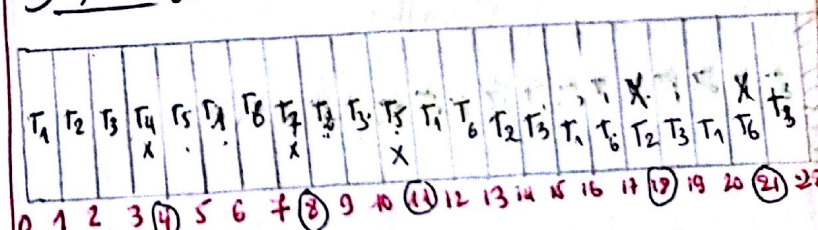


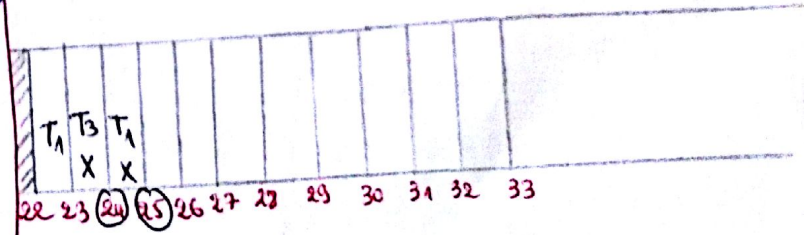
- temps moyen :

$$tm = \frac{8 + (2-1) + (3-2) + (5-1) + (12-2) + (18-1) + (25-0)}{7}$$

$tm = 9,43$

4) Quantum de 1 :

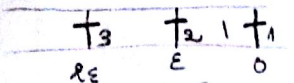




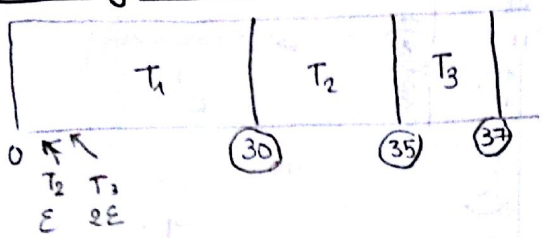
Exo n° 1 : (1ère partie) TD N° 4 :

FIFO : First In First Out.

Premier arrivé premier servie.



1) Diagramme de Gantt :



2) Temps moyen d'exécution :

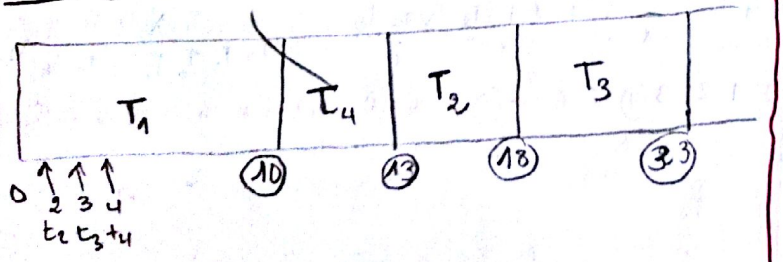
$$t_m = \frac{(30-0) + (35-\varepsilon) + (37-2\varepsilon)}{3}$$

$$t_m = \frac{102 - 3\varepsilon}{3}.$$

Exo n°2 : PCTE

Plus court temps d'exécution qui s'exécute le premier.

1) Diagramme de Gantt :



T_1 en premier, car en temps 0 il y'a que T_1 qui est chargé en mémoire, donc il s'exécute le premier, puis en temps 10, les 3 autres sont chargés en mémoire, donc, on calcule celui qui a le plus court temps d'exécution, et c'est elle qui va s'exécutée.

2) temps moyen :

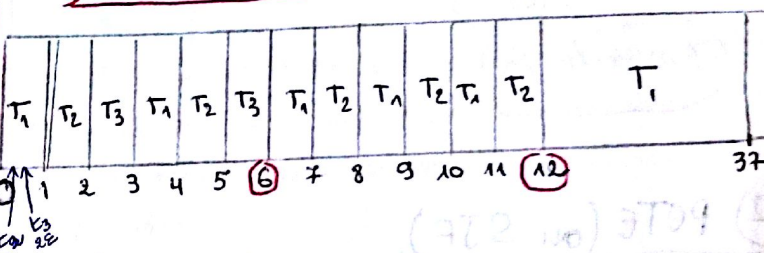
$$t_m = \frac{10 + (13-4) + (18-2) + (33-3)}{4}$$

$$t_m = \frac{64}{4} = 16,25.$$

Exo 3 : (Tourniquet)

I) Diagramme de Gantt :

I, 1) Quantum = 1 :

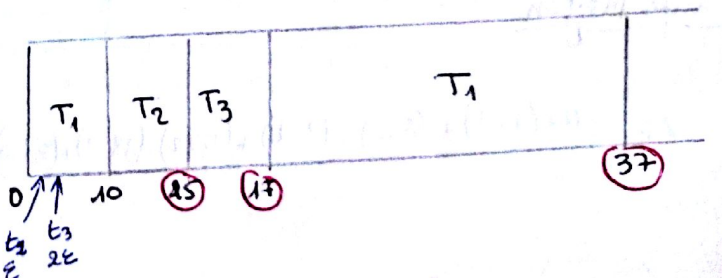


II. 1) temps moyen :

$$t_m = \frac{37 + (6-1\varepsilon) + (12-\varepsilon)}{3}$$

$$t_m = 18,33 - \varepsilon.$$

I, 2) Quantum = 10 :

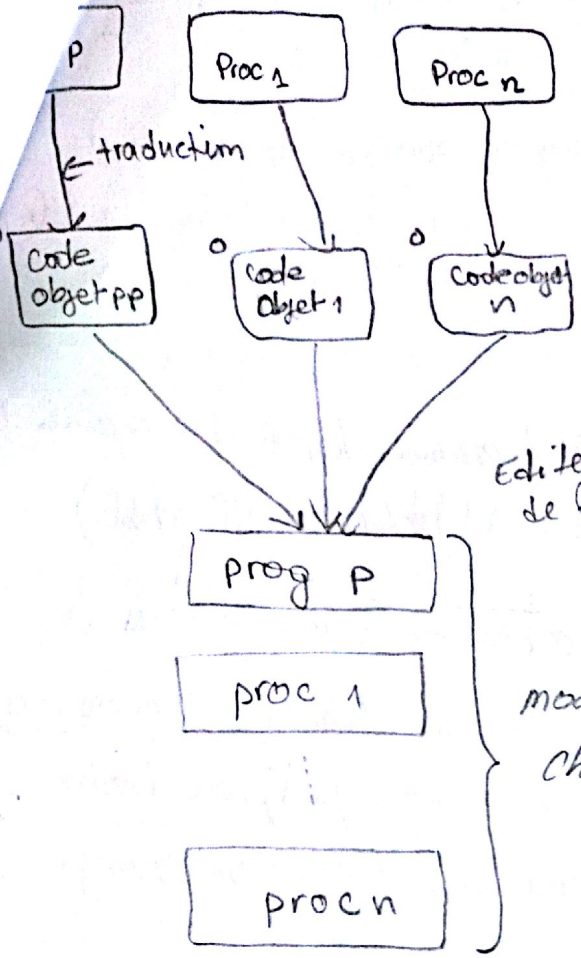


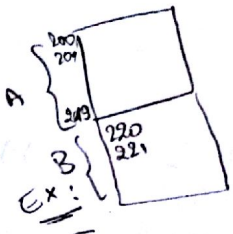
Temps moyen :

$$t_m = \frac{37 + (15-\varepsilon) + (17-2\varepsilon)}{3}$$

$$t_m = 23 - \varepsilon$$

SERIE TD N°3





on a cte de translation 200 càd : on ajoute 200 à chaque fois.

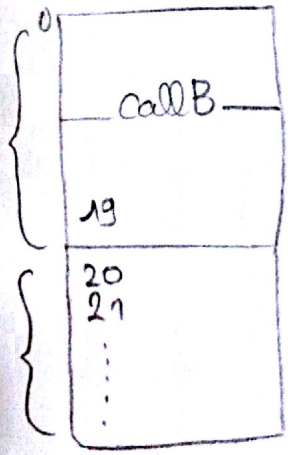
chargement (En mémoire)

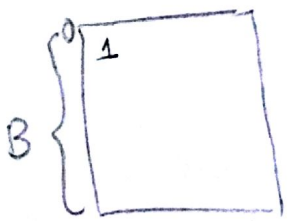
chargeur absolu

chargeur relogeable

On utilise un registre de Base qui va contenir la cte de translation (200)

@ effective = @ de base + déplacement





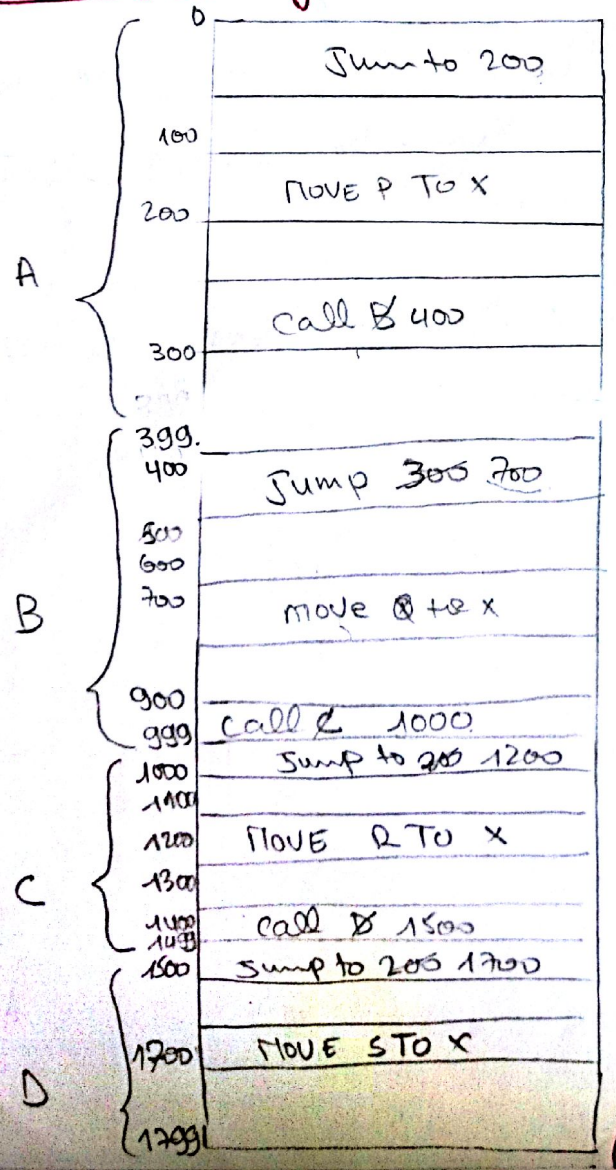
Exo

1- création du module chargeable

table des modules :

Nom	Adresse début	longueur	adresse fin
A	0	400 (0 → 399)	399
B	400 adresse fin A + 1	600	999 adresse fin A + longueur B
C	1000	500	1499
D	1500	300	1799

le module chargeable : (*)



2- Le contenue du module chargé

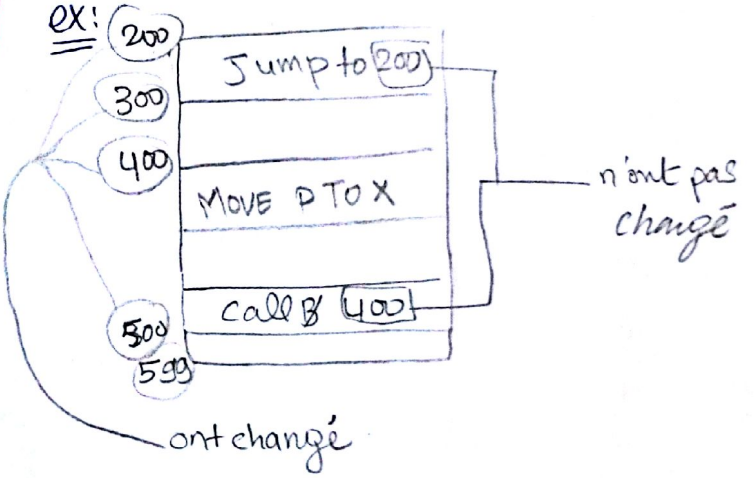
1er cas : JL n'existe pas de registre de (càd : chargeur absolue).

on ajoute 200, (@ d'implantation (Cte de translation) a chaque adresse du module chargeable (*)

2ème cas : La machine dispose d'un registre de base (càd chargeur relogeable).

dans ce cas, on ne modifie pas les adresse qui sont dedans (ex Jump to 200) Le 200 ne change pas, par contre les @ qui sont dehors vont changé

ex :



$T_2\ T_1$

$T_1\ T_5\ T_4\ T_3\ T_2$

$T_2\ T_7\ T_6\ T_1\ T_5\ T_4\ T_3$ ← ordre final.

$t = 3$ $T_3\ T_2\ T_7\ T_6\ T_1\ T_5\ T_4$

$t = 4$ $T_4\ T_3\ T_2\ T_7\ T_6\ T_1\ T_5$

⋮

$$t_m = \frac{25 + (4-1) + (8-2) + (11-1) + (21-2) + (18) + (24-1)}{7}$$

$t_m = 14,86$.