

TD N°1 Système d'exploitation 2

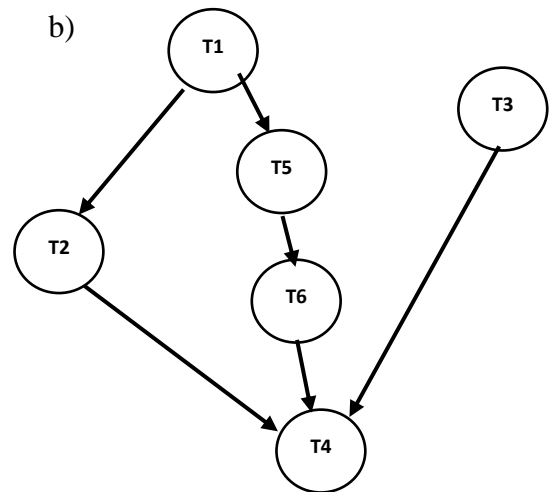
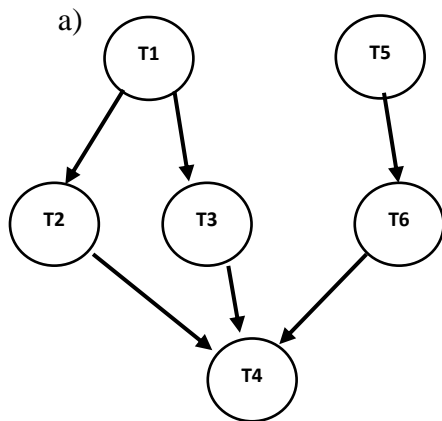
Exo.1

Donner les graphes de précedence correspondants aux notations suivantes :

- a) $(T1.T2) // T3$
- b) $(T1.T3) // (T2.T4).T5$
- c) $T1.(T2 // T3) // (T4.T5)$

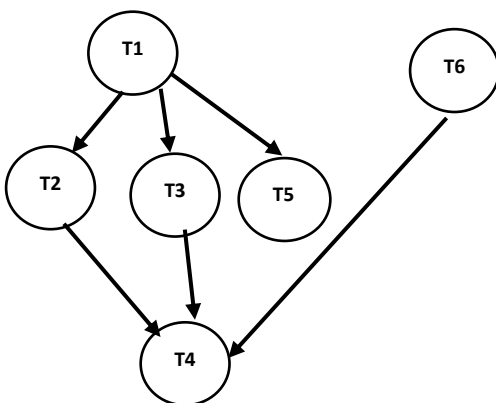
Exo.2

A partir des graphes suivant donner les notations correspondantes ainsi que l'algorithme en Parbegin, Parend



Exo.3

A partir du graphes suivant donner l'algorithme en Fork et Join (goto)



Exo.4

On considère le programme utilisant les primitives parbegin/parend donné ci-dessous:

```
begin
  parbegin
    lire(a)
    lire(b)
  parend ;
  parbegin
    c := a*a
    begin
      d := a*b ;
      e := d*a ;
    end
  parend ;
  e := c + d + e
end
```

- Donner le graphe de précedence en précisant les tâches que vous considérez
- Indiquer les domaines de lecture et écriture des différentes tâches
-

Exo.5

Vérifier la possibilité de parallélisations (condition de Bernstein) des tâches suivantes :

$$T1 : x_1 = x + y$$

$$T4 : x_4 = x_2 * x_3$$

$$T2 : x_2 = 2z$$

$$T5 : x_5 = x_1 - x_4$$

$$T3 : x_3 = x / y$$

- Dessiner le graphe de précedence
- Ecrire les algorithmes en ParBegin/ParEnd, puis en Fork/Join

Exo.6

Soit les équations suivantes :

$$Z = (a+b).(c-d) + e.f$$

$$T = 2z + (a+b).(c-d)$$

$$G = T + Z$$

$$U = 2G + T$$

- Décomposer les équations en tâches élémentaires
- Quelles tâches peuvent être parallélisé ?
- Dessiner le graphe de précedence correspondant