

IFA-3-SYS : Examen du 19/03/2025

Q1. flags attendus : zxf. Le reste est facultatif

Q2. touch fichierA
mkdir dossier1
mv fichierA dossier1
touch fichierB
mkdir dossier2
mv fichierB dossier2
touch fichierC
mkdir dossier4
mv fichierC dossier4
mkdir dossier3
mv dossier4 dossier3

Q3. Vrai ; Faux ; Faux ; Vrai

Q4. objdump

Q5. Faux (chacun sa pile) ; Vrai (le même espace d'adressage virtuel) ; Faux (pas nécessairement) ; Faux (pas spécialement)

Q6. NEW --> READY ----> RUNNING --> TERMINATED

```
  /|\ /|\      | |
  |   \___P___/ |
  \           |
  \_ BLOCKED <---/
```

Q7. Process Control Block

Q8. time 0 20 40 60 80 90 100 150 170 190 210
CPU | B | E | B | A | D | G | A | C | F | C |

Q9. 189 ; EA5 ; FA5 ; 91F. (Les numéros de page sont sur 3 bits)

Q10. il est BLOQUÉ : l'ordonnanceur ne distingue pas une faute de page VS une requête d'IO volontaire

Q11. Faux ; Faux ; Faux ; Vrai

Q12. X dans C (reste 10 kio), puis Y dans D (reste 6 kio) puis Z dans C (reste 1 kio)

Q13. mutex=NON. Exemple : A1 A2 B1 B2 A4 B4

Q14. deadlock-freedom=OUI

Preuve par l'absurde. Le seul endroit possible pour un interblocage est la paire de boucles A2-A3-B2-B3. Si les deux threads sont là-dedans, alors `locked==true`, pourtant la dernière affectation dans le passé (que ce soit chez A ou chez B) disait `locked=false` $\Rightarrow$ contradiction.

non-interference=OUI

Si A s'arrête dans `remainder()` alors B peut faire des tours de sa boucle, et respectivement.

Q15. mutex=OUI

Preuve par l'absurde : supposons A5 et B5. Alors $turn == 0$, sinon A serait encore en A2-A3. Mais aussi $turn == 1$ sinon V serait encore en B2-B3. $\Rightarrow$ contradiction

Q16. Progression=NON car interférence.

Preuve : supposons A1...A9 puis A s'arrête dans le remainder $\Rightarrow$ B bloqué en B2-B3

Q17. mutex=OUI

Preuve par l'absurde : supposons A6 et B6. Alors $flag = \{true, true\}$ (à cause de A2-A8 et B2-B8). Mais B6 implique que $flag[0] == false$ (sinon on serait en B3-B4) $\Rightarrow$ contradiction

Q18. progression=NON car deadlock possible si A2 A3 B2 B3 (aka «excès de courtoisie»)

Q19. mutex=OUI

Preuve idem Q17 : supposons A7 et B7. Alors $flag[0] == true$ (car A7 est entre A2 et A9) et de même $flag[1] == true$.

Or les deux threads ont chacun pu dépasser leur boucle while : donc soit $flag[0] == false$ (non), soit $flag[1] == false$ (non plus) soit $turn \notin \{0,1\}$ (contradiction).

Q20. deadlock-free=OUI

Supposons un deadlock, i.e. A4-A5 et B4-B5. alors $turn \notin \{0,1\}$ (contradiction).

non-interference=OUI

Si un thread s'arrête dans son remainder alors son flag est false et il ne peut pas interférer avec l'autre.