

### IF-3-SYS : Contrôle continu. Avril 2019.

**Q1.** Vrai (par définition de VCPU) ; Faux (cf cours 1, les instructions sont exécutées par le matériel) ; Faux (une IRQ matérielle, plutôt) ; Vrai (par définition de «trappe»)

**Q2.** Faux (il aurait le droit) ; Vrai (le processus enfant n'a pas changé son programme) ; Faux (le noyau unix ne propose pas ce service) ; Faux (il faudrait pour ça que le parent fasse un `wait()` explicite)

**Q3.** Quatre fois (personne ne répète la boucle while)

**Q4.** `read`, `sleep`, `waitpid`. Rappel : une fonction est dite *bloquante* ssi appeler cette fonction peut me faire passer dans l'état BLOCKED.

**Q5.** 1) rien, car il attend que le matériel termine une opération de lecture/écriture. 2) une interruption (IRQ) en provenance du périphérique concerné

**Q6.** Seulement Running. (par définition du multitâche, on peut avoir autant de processus que l'on veut)

**Q7.**

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

**Q8.**  $AMAT = 10ns + 1\% \times 100ns = 11ns$

**Q9.**

- Chaque traduction conserve 12 bits d'offset, on en déduit que la taille des pages est  $2^{12} = 4$  kio.
- Les adresses sont exprimées sur 16 bits, donc la taille max d'un VAS est  $2^{16} = 64$  kio.
- Pour la taille de RAM on ne peut pas conclure : le PAS fait visiblement 64 kio aussi, mais on ne peut pas savoir quelle proportion correspond à de la RAM.

**Q10.** MMU, processeur, suspendu, table de pagination.

**Q11.** Faux et Faux (allocation statique = adresses (virtuelles) écrites en dur dans l'exécutable) ; Faux et Vrai (la pile contient les variables locales et les adresses de retour des fonctions, mais pas de code)

**Q12.** Vrai Vrai Faux Faux. La pile et le tas ne figurent pas dans l'exécutable ; ils sont vides au début de l'exécution, et sont alloués peu à peu par le noyau.

**Q13.** X dans C (reste 10 kio), puis Y dans D (reste 6 kio) puis Z dans C (reste 1 kio)