

**NOM Prénom :**

### Consignes

- Durée : 30 minutes. Lisez le sujet en entier (10 questions sur 4 pages) avant de commencer.
- Écrivez lisiblement et surtout sans ratures. Utilisez un brouillon (vraiment).
- Les réponses seront à inscrire sur le sujet. Commencez par écrire votre nom ci-dessus.
- Documents et appareils interdits, sauf une feuille A4 recto-verso manuscrite.
- Pour les calculs en binaire, vous pouvez vous aider des tableaux donnés en page 4.
- Dans les questions vrai/faux, les erreurs sont décomptées : ne répondez pas au hasard.

**Question 1** Pour chaque acronyme ci-dessous, donnez sa signification en toutes lettres :

INSA	Institut National des Sciences Appliquées
CLI	
FCFS	
ISR	
VAS	

## 1 Noyau et processus

**Question 2** Pour chaque proposition ci-dessous, entourez V si elle est correcte, ou entourez F si elle est fausse ou absurde.

- F Comme les processus utilisateur s'exécutent en «mode restreint», ils ne peuvent pas changer le contenu des registres du processeur.
- F Le rôle du noyau est, entre autres, d'interpréter les commandes tapées par l'utilisateur dans le terminal.
- F Sous Ubuntu, Fedora, et Android, les appels système disponibles sont les mêmes car ces trois systèmes sont tous basés sur le noyau Linux.
- F Une trappe est une interruption causée volontairement par un programme plutôt que par un évènement matériel.

**Question 3** Dans cette question, on suppose l'existence de deux fonctions permettant de convertir des nombres d'une représentation à l'autre : int atoi(char \* str) et char \* itoa(int val). On compile ce programme en un exécutable prog qu'on lance en tapant ./prog 5. Combien de fois la lettre Z sera-t-elle affichée ?

Ce programme affichera Z  fois au total.

```
int main(int argc, char * argv[])
{
    assert(argc==2);
    int N=atoi(argv[1]);
    if(N)
    {
        fork();
        char * param=itoa(N-1);
        exec(argv[0], param);
        print('Z');
    }
    else
    {
        print('Z');
    }
}
```

**Question 4** De quoi parle-t-on quand on dit qu'un appel système est bloquant ? Donnez une définition en une seule phrase. Illustrez par un exemple d'appel bloquant et un exemple d'appel non bloquant.

---



---



---



---



---

## 2 Ordonnancement processeur

**Question 5** Pour chaque proposition ci-dessous, entourez V si elle est correcte, ou entourez F si elle est fausse ou absurde.

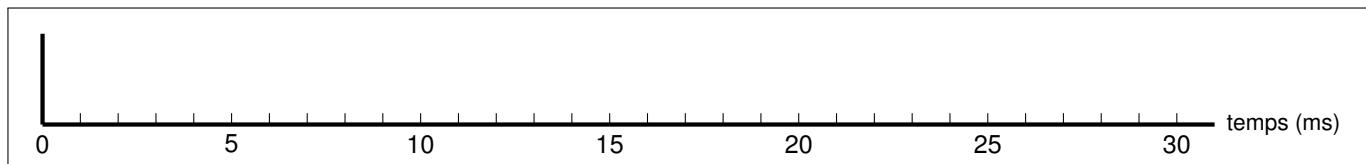
- F Avec un ordonnanceur préemptif, par définition, il ne peut jamais y avoir de famine.
- F Lors d'un appel à `fork()`, le nouveau processus est toujours placé dans la *Ready Queue*.
- F Si un machine possède plusieurs CPU, alors il faut plusieurs ordonnanceurs.
- F Un programme interactif aura en général un comportement de type «CPU-bound».

**Question 6** On s'intéresse dans cette question à trois processus décrits dans le tableau ci-dessous. L'activité de chaque processus consiste en une phase de calcul (CPU burst), suivie d'une phase d'entrées-sorties (IO burst) puis d'une seconde phase de calcul. À l'instant initial, le processus A est prêt à s'exécuter. Les autres instants d'arrivée sont indiqués dans le tableau. Tous les temps sont en millisecondes.

Processus	A	B	C
Instant d'arrivée	0	3	5
1 <sup>ière</sup> CPU-burst,	4	6	3
puis IO-burst	4	4	7
puis 2 <sup>nde</sup> CPU-burst	7	6	2

On suppose que la machine ne dispose que d'un unique processeur, ordonné selon la stratégie SRTF. Par contre elle peut faire simultanément plusieurs opérations d'entrées-sorties.

Au brouillon, dessinez un chronogramme indiquant la succession des tâches sur le processeur. Recopiez ensuite votre réponse au propre dans le cadre ci-dessous.



**Question 7** On s'intéresse dans cette question à un système multitâche dans lequel les processus calculent en moyenne pendant une durée  $T$  entre deux requêtes d'entrées-sorties. Chaque fois qu'un processus se bloque, l'ordonnanceur bascule vers un processus prêt (on suppose qu'il y en a toujours). On note  $S$  ce temps de *context-switch* passé à exécuter du code noyau. L'ordonnanceur applique une politique round-robin avec un quantum de durée  $Q$ . On suppose que  $S < Q < T$ .

On souhaite évaluer le taux d'utilisation du processeur c'est à dire la fraction du temps effectivement passé à exécuter du code applicatif. Donnez une formule pour  $U$  en fonction des différents paramètres du modèle :

### 3 Mémoire virtuelle

**Question 8** Pour chaque proposition ci-dessous, entourez V si elle est correcte, ou entourez F si elle est fausse ou absurde.

- Si on invoque `mmap()` avec l'option `MAP_FILE`, alors il faut avoir invoqué préalablement `open()` pour ouvrir le fichier.
- Suivant les options qu'on lui passe, l'appel système `mmap()` peut modifier notre table de pagination.
- Suivant les options qu'on lui passe, l'appel système `mmap()` peut nous permettre de lire et d'écrire dans un fichier.
- Suivant les options qu'on lui passe, l'appel système `mmap()` peut nous retourner une adresse virtuelle ou une adresse physique.

**Question 9** Complétez ce paragraphe avec les termes manquants. Un *défaut de page* est détecté

par  lorsque le programme essaye d'accéder à une de ses pages, mais que celle-ci n'est pas présente en mémoire. En réaction,  saute alors vers le noyau. Celui-ci envoie au disque dur un ordre de lecture pour charger vers la mémoire les

données manquantes. Le processus en question est marqué comme

et l'ordonnanceur en choisit un autre à exécuter. Lorsque l'opération de lecture est terminée, le noyau met à jour  du processus d'origine, et le marque à nouveau comme

**Question 10** On suppose dans cette question un système avec mémoire virtuelle paginée. Les adresses (virtuelles et physiques) sont exprimées sur 8 bits, et la taille des pages (virtuelles et physiques) est de 32 octets.

Soit un processus dont la table de page est la suivante. Le signe  $\emptyset$  désigne un PTE invalide.

VPN	0	1	2	3	4	5	6	7
PPN	$\emptyset$	7	3	$\emptyset$	2	$\emptyset$	$\emptyset$	$\emptyset$

Quelle adresse physique correspond à l'adresse virtuelle 0x55 ? Répondez en hexadécimal.

## Annexe : aide pour les calculs en binaire

Les premiers nombres entiers, notés en décimal, hexadécimal, et binaire :

Dec	Hex	Bin	Dec	Hex	Bin	Dec	Hex	Bin	Dec	Hex	Bin
0	0	0	5	5	101	10	A	1010	15	F	1111
1	1	1	6	6	110	11	B	1011	16	10	10000
2	2	10	7	7	111	12	C	1100	17	11	10001
3	3	11	8	8	1000	13	D	1101	18	12	10010
4	4	100	9	9	1001	14	E	1110	19	13	10011

Les premières puissances de 2, notées en décimal :

$2^0 =$	1	$2^{16} =$	65 536	$2^{32} =$	4 294 967 296	$2^{48} =$	281 474 976 710 656
$2^1 =$	2	$2^{17} =$	131 072	$2^{33} =$	8 589 934 592	$2^{49} =$	562 949 953 421 312
$2^2 =$	4	$2^{18} =$	262 144	$2^{34} =$	17 179 869 184	$2^{50} =$	1 125 899 906 842 624
$2^3 =$	8	$2^{19} =$	524 288	$2^{35} =$	34 359 738 368	$2^{51} =$	2 251 799 813 685 248
$2^4 =$	16	$2^{20} =$	1 048 576	$2^{36} =$	68 719 476 736	$2^{52} =$	4 503 599 627 370 496
$2^5 =$	32	$2^{21} =$	2 097 152	$2^{37} =$	137 438 953 472	$2^{53} =$	9 007 199 254 740 992
$2^6 =$	64	$2^{22} =$	4 194 304	$2^{38} =$	274 877 906 944	$2^{54} =$	18 014 398 509 481 984
$2^7 =$	128	$2^{23} =$	8 388 608	$2^{39} =$	549 755 813 888	$2^{55} =$	36 028 797 018 963 968
$2^8 =$	256	$2^{24} =$	16 777 216	$2^{40} =$	1 099 511 627 776	$2^{56} =$	72 057 594 037 927 936
$2^9 =$	512	$2^{25} =$	33 554 432	$2^{41} =$	2 199 023 255 552	$2^{57} =$	144 115 188 075 855 488
$2^{10} =$	1 024	$2^{26} =$	67 108 864	$2^{42} =$	4 398 046 511 104	$2^{58} =$	288 230 376 151 711 744
$2^{11} =$	2 048	$2^{27} =$	134 217 728	$2^{43} =$	8 796 093 022 208	$2^{59} =$	576 460 752 303 423 488
$2^{12} =$	4 096	$2^{28} =$	268 435 456	$2^{44} =$	17 592 186 044 416	$2^{60} =$	1 152 921 504 606 846 976
$2^{13} =$	8 192	$2^{29} =$	536 870 912	$2^{45} =$	35 184 372 088 832	$2^{61} =$	2 305 843 009 213 693 952
$2^{14} =$	16 384	$2^{30} =$	1 073 741 824	$2^{46} =$	70 368 744 177 664	$2^{62} =$	4 611 686 018 427 387 904
$2^{15} =$	32 768	$2^{31} =$	2 147 483 648	$2^{47} =$	140 737 488 355 328	$2^{63} =$	9 223 372 036 854 775 808
						$2^{64} =$	18 446 744 073 709 551 616

On rappelle également que :

- 1 kio = 1024 octets,
- 1 Mio = 1024 Kio,
- 1 Gio = 1024 Mio,
- 1 Tio = 1024 Gio,
- etc. (avec dans l'ordre : Pio, Eio, Zio, Yio)

En cas de doute sur ces unités, n'hésitez pas à demander des précisions.