

NOM Prénom :

Consignes

- Durée : 30 minutes. Lisez le sujet en entier (9 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.

1 Noyau et processus

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

INSA	Institut National des Sciences Appliquées
DMA	
IRQ	
PCB	
VAS	

Question 2 Pour chaque proposition ci-dessous, entourez V s'il s'agit d'un évènement qui provoque une exception/interruption¹ ou entourez F dans le cas contraire.

- | | | |
|----------------------------|----------------------------|---|
| <input type="checkbox"/> V | <input type="checkbox"/> F | L'utilisateur presse une touche du clavier. |
| <input type="checkbox"/> V | <input type="checkbox"/> F | Le noyau exécute une instruction privilégiée. |
| <input type="checkbox"/> V | <input type="checkbox"/> F | Une application fait un appel système. |
| <input type="checkbox"/> V | <input type="checkbox"/> F | Une application fait une division par zéro. |

1. Dans cette question on ne fait pas de distinction particulière entre les deux termes

Question 3 On s'intéresse dans cette question aux deux programmes ci-dessous², qu'on suppose déjà compilés en deux exécutables `./main` et `./multiecho`.

main.c

```
main(int argc, char** argv)
{
    int r = fork();
    if(r == 0)
    {
        exec("./multiecho","4","A");
    }
    else
    {
        exec("./multiecho","3","B");
    }
    wait(NULL);
    wait(NULL);
    printf("C\n");
}
```

multiecho.c

```
int main(int argc, char** argv)
{
    assert(argc == 3);
    int num = atoi(argv[1]);
    assert(num > 0);
    for(i=0; i<num; i++)
    {
        printf("%s\n",argv[2]);
    }
    return 0;
}
```

Sur la ligne de commande, l'utilisateur tape `./main`. Pour chaque proposition ci-dessous, entourez V si cet affichage est possible (on ignore les retours à la ligne) ou entourez F sinon.

- | | | |
|---|---|----------|
| V | F | AAAABBBC |
| V | F | ABABABA |
| V | F | BBBAAAA |
| V | F | CABABABA |

2 Ordonnancement processeur

Question 4 Pour chaque proposition ci-dessous, entourez V si elle correspond à une transition possible entre deux états d'un processus, ou entourez F dans le cas contraire.

- | | | |
|---|---|-------------------|
| V | F | READY → BLOCKED |
| V | F | READY → RUNNING |
| V | F | RUNNING → BLOCKED |
| V | F | RUNNING → READY |

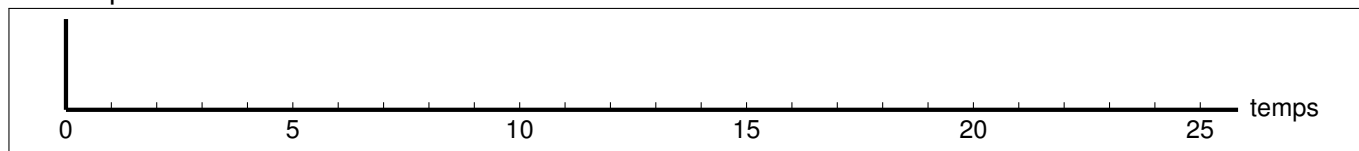
Question 5 Dans le cours sur l'ordonnancement on a beaucoup utilisé le terme de *CPU burst*. De quoi s'agit-il ? Quelle est la différence avec un programme ? avec un processus ? Rédigez votre réponse au brouillon avant de la recopier ci-dessous.

2. Par souci de lisibilité, on utilise ici une simplification (imaginaire) de l'appel système `exec()` avec seulement les arguments de ligne de commande, sans répétition. Avec la syntaxe réelle, le premier appel s'écrirait par exemple `execl("./multiecho", "./multiecho", "4", "A", NULL)`; mais ça ne changerait rien à la consigne de l'exercice.

Question 6 On s'intéresse dans cette question à trois tâches avec les caractéristiques suivantes :

nom	A	B	C
instant d'arrivée	0	2	4
durée d'exécution	8	4	8

On suppose que l'ordonnanceur applique la stratégie Round Robin avec un quantum de 3 unités de temps. La durée des changements de contexte est supposée négligeable. Sur une feuille de brouillon, dessinez un chronogramme indiquant la succession des tâches sur le processeur. Recopiez ensuite votre réponse dans le cadre ci-dessous.



3 Mémoire virtuelle

Question 7 Indiquez les termes manquants dans le paragraphe ci-dessous.

Grâce à la mémoire virtuelle, le système donne l'illusion aux de disposer d'une mémoire plus que celle réellement présente sur la machine. Cependant, cette mémoire leur apparaîtra aussi plus à cause des qui se produiront de temps en temps.

Question 8 On s'intéresse dans cette question à un système avec mémoire virtuelle où les adresses virtuelles sont codées sur 32 bits, avec des pages de 4Kio. Combien de bits sont alors nécessaires pour encoder un numéro de page virtuelle ? Pour les calculs, vous pouvez vous aider du tableau page suivante.

Question 9 Pour demander qu'on lui alloue une nouvelle région de mémoire, un processus peut invoquer l'appel système `mmap()`. Mais dans certaines circonstances, le noyau ne parviendra pas satisfaire cette demande. Pour chaque proposition ci-dessous, indiquez si elle est plausible (entourez V), ou bien si elle est fausse, hors-sujet et/ou absurde (entourez F).

- ☐ V ☐ F Un `mmap()` peut échouer par manque de place libre dans la mémoire vive de la machine.
- ☐ V ☐ F Un `mmap()` peut échouer par manque de place libre dans l'espace d'adressage du processus.
- ☐ V ☐ F Un `mmap()` peut échouer par manque de place libre sur le disque dur.
- ☐ V ☐ F Un `mmap()` peut échouer par manque de place libre dans le noyau.

Annexe : aide pour les calculs en binaire

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

Dec	Hex	Bin
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100

Dec	Hex	Bin
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001

Dec	Hex	Bin
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110

Dec	Hex	Bin
15	F	1111
16	10	10000
17	11	10001
18	12	10010
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.