TELECOM nancy

Examen RSA du 23/04/2021

RSA : Réseaux et Systèmes Avancés

Deuxième année

UNIVERSITÉ DE LORRAINE

Épreuve individuelle sans documents. Seule une feuille a4 de notes personnelles manuscrites est autorisée. Le barème est donné à titre indicatif. Durée 1h30. Répondre et inscrire nom et prénom au verso. Rendre juste la présente page.

★ **Exercice 1: Table de dispatching (Solaris) (3 pts)** Soit une vue partielle d'une table de dispatching (type Solaris). Relever les éventuelles incohérences. Justifier.

priority	time quantum	new priority when time quantum expires	new priority when return from sleep
0	200	10	0
30	80	35	20
40	40	45	30
55	40	60	40

★ **Exercice 2: QCM (17 pts)** (+1 par case correctement cochée, −0.25 sinon)
Cocher les bonnes réponses :

1. Dans un Linux 2.6, le nombre de thread maximal
   — a une borne
   ☐ inférieure ☐ supérieure
   — dépend de la taille de la mémoire
   ☐ virtuelle ☐ physique
2. Le nombre de threads par processus Linux est :
   ☐ donné par cat /proc/sys/kernel/threads-max
   ☐ 1
   ☐ limité par la taille de la mémoire physique disponible (nombre de cases/4)
   ☐ 30709
3. Concernant l'ordonnanceur Linux 2.6,
   ☐ 99 niveaux de priorité temps réel sont possibles
   ☐ 40 niveaux de priorité conventionnelle sont possibles
   ☐ on utilise un bitmap au même titre que la runqueue utilisée dans SOLARIS
   ☐ la valeur de nice d'un processus détermine sa priorité dynamique
   ☐ le timeslice est calculé sur la base de la priorité dynamique d'un processus
4. Dans un ordonnancement type *round-robin*,
   un grand quantum permet une bonne utilisation :
   ☐ des périphériques d'E/S
   ☐ du processeur ;
   un petit quantum permet une bonne utilisation :
   ☐ des périphériques d'E/S
   ☐ du processeur

★ **Exercice 3: Pagination (30 pts)**
Soit une machine 32 bits. La table au verso donne le contenu des entrées de la table des pages (TPE) dont le bit de présence est positionné à 1.

▷ **Question 1:** En supposant l'utilisation d'un seul niveau de pagination et une taille de page de 4 Ko, quel est le format d'une adresse virtuelle ?

NOM : Prénom :

i (décimal)	1	2	3	7	9	15	150	167	250	265	325	405	602	2800
$TPE[i].case$ (hexa)	ffff	8	804	a25	bf	10a	12a	110	20b	65a	3b2	3f5	7ffff	eb9

▷ **Question 2:** En déduire la taille (en octets sous la forme $x \times 2^y$) de la table des pages associée à chaque processus. Une entrée de cette table fait 4 octets.

▷ **Question 3:** Remplir le tableau ci-dessous en donnant le résultat de la conversion des adresses logiques de la première ligne.

@ logique	0x 9613	0x fab0	0x af001	0x 3550	0x 25adc0
@ physique	0x	0x	0x	0x	0x

▷ **Question 4:** Toujours avec un seul niveau de pagination mais une taille de page de 256 octets, quel est le format d'une adresse virtuelle ?

▷ **Question 5:** Donner le résultat de la conversion des adresses logiques en considérant une taille de la page est de 256 octets :

@ logique	0x 9613	0x fab0	0x af001	0x 3550	0x 25adc0
@ physique	0x	0x	0x	0x	0x

▷ **Question 6:** On suppose maintenant que l'on utilise une pagination à 3 niveaux. La PMD et la PGD comportent 256 et 4096 entrées respectivement. La taille d'une case mémoire est de 256 octets. Donner le format de l'adresse virtuelle.

★ Exercice 4: Mémoire virtuelle d'un processus (24 pts)

Soit le programme C du listing 1 dont le résultat partiel de son exécution est donné par le listing 2. La taille d'une page = 4 Ko

▷ **Question 1:** Le listing 3 donne la sortie d'une commande mystère à la suite de l'exécution du programme. Quelle est cette commande ?

▷ **Question 2:** Le listing 4 donne la sortie d'une autre commande mystère. Quelle est cette commande ?

▷ **Question 3:** Compléter le tableau suivant.

Variable	Adresse possible	Région mémoire d'appartenance
px[0]	0x	
tab1[0]	0x	
tab2[0]	0x	

▷ **Question 4:** S'agit-il d'adresses physiques ou logiques ?

▷ **Question 5:** Donner la taille de tab2 en pages

▷ **Question 6:** Donner la taille de la pile en pages

▷ **Question 7:** Quelle est la taille minimale du bss ? (nombre d'octets en décimal sous la forme $x \times 2^y$)

NOM : Prénom :

Listing 1 – Programme C

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

#define NB_PAGES (1<<10)
#define PAGE_SIZE (1<<12)
#define MAX (NB_PAGES*PAGE_SIZE)

char tab1[MAX];

int main (void){

  int i;

  char tab2[MAX];
  int  *px = malloc (MAX*sizeof(int));

  for (i=0; i<MAX; i++){
    tab1[i] = 'x';
    tab2[i] = 'y';
    px[i] = i;
  }

  printf ("l'adresse de  px[0] = %010p\n", &px[0]);
  printf ("l'adresse de  tab1[0] = %010p\n", &tab1[0]);
  printf ("l'adresse de  tab2[0] = %010p\n", &tab2[0]);

  printf ("memory map file: /proc/%d/maps\n", getpid());
  printf ("sleeping ...");
  fflush (NULL);
  sleep (20);

  free (px);

  exit (EXIT_SUCCESS);
}
```

Listing 2 – Résultat partiel de l'exécution du programme C

```
l'adresse de  px[0] = 0x ............
l'adresse de  tab1[0] = 0x ..............
l'adresse de  tab2[0] = 0x ..............
memory map file: /proc/14754/maps
sleeping ...
```

Listing 3 – Sortie de la commande mystère 1

```
08048000-08049000 r-xp 00000000 08:06 1588617    ~/2-tp/files/mem
08049000-0804a000 rw-p 00000000 08:06 1588617    ~/2-tp/files/mem
0804a000-0844a000 rw-p 00000000 00:00 0
b6659000-b765b000 rw-p 00000000 00:00 0
b765b000-b77a7000 r-xp 00000000 08:04 1835021    /lib/i386-linux-gnu/libc-2.13.so
b77a7000-b77a9000 r--p 0014c000 08:04 1835021    /lib/i386-linux-gnu/libc-2.13.so
b77a9000-b77aa000 rw-p 0014e000 08:04 1835021    /lib/i386-linux-gnu/libc-2.13.so
b77aa000-b77ad000 rw-p 00000000 00:00 0
b77c3000-b77c6000 rw-p 00000000 00:00 0
b77c6000-b77c7000 r-xp 00000000 00:00 0          [vdso]
b77c7000-b77e3000 r-xp 00000000 08:04 1835014    /lib/i386-linux-gnu/ld-2.13.so
b77e3000-b77e4000 r--p 0001b000 08:04 1835014    /lib/i386-linux-gnu/ld-2.13.so
b77e4000-b77e5000 rw-p 0001c000 08:04 1835014    /lib/i386-linux-gnu/ld-2.13.so
bf6dc000-bfadf000 rw-p 00000000 00:00 0          [stack]
```

Listing 4 – Sortie de la commande mystère 2

```
text    data     bss     dec     hex filename
2123    320    4194336 4196779 4009ab mem
```