Université de Reims
Champagne-Ardenne
U.F.R. de Sciences
Exactes et Naturelles

Licence MMI
Info0403
2020/2021

Devoir Surveillé Terminal

Session 1 – 1 h30

Resp. Th. Bernard

Exercice 1 (Cours (3 pts))

- Explicitez les spécificités de la deuxième génération d'ordinateurs 1955-1965 (1pt).
- Expliquez le mécanisme de pagination (1pt).
- Que fait la fonction *signal*() (1pt) ?

Exercice 2 (Va et Vient (4 pts))

On considère le mécanisme de va et viens avec la série de chargement/déchargement suivante :

1. Chargement processus A de 3 UA,
2. Chargement processus B de 7 UA,
3. Chargement processus C de 2 UA,
4. Déchargement processus A,
5. Chargement processus D de 4 UA,
6. Chargement processus E de 5 UA,
7. Chargement processus F de 3 UA,
8. Déchargement processus D,
9. Chargement processus A de 3 UA,
10. Déchargement processus F,
11. Chargement processus G de 4 UA,
12. Déchargement processus E,
13. Chargement processus F de 3 UA.

En considérant une mémoire pouvant contenir 24 unités d'allocation, donner l'état de la mémoire à chaque étape en utilisant les stratégies suivantes :

1. First Fit

1																							
2																							
3																							
4																							
5																							
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							

2. Next Fit

1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						
11																						
12																						
13																						

3. Best Fit

1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						
11																						
12																						
13																						

4. Worst Fit

1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						
11																						
12																						
13																						

Exercice 3 (Pagination (5 pts))
On considere la séquence de pages suivante :

$$\sigma = 1, 2, 3, 4, 3, 1, 2, 4, 5, 3, 1, 3, 4, 2, 1, 6, 5, 6, 7, 2, 5, 3, 4, 1, 2, 4, 7$$

On dispose d'une mémoire rapide contenant un emplacement pour 4 pages.

Calculez le nombre de défaut de pages pour chacun des algorithmes suivants :

— Belady

1	2	3	4	3	1	2	4	5	3	1	3	4	2	1	6	5	6	7	2	5	3	4	1	2	4	7

— First In First Out

1	2	3	4	3	1	2	4	5	3	1	3	4	2	1	6	5	6	7	2	5	3	4	1	2	4	7

— Last In First Out

1	2	3	4	3	1	2	4	5	3	1	3	4	2	1	6	5	6	7	2	5	3	4	1	2	4	7

— Least Recently Used

1	2	3	4	3	1	2	4	5	3	1	3	4	2	1	6	5	6	7	2	5	3	4	1	2	4	7

— Flush When Full

1	2	3	4	3	1	2	4	5	3	1	3	4	2	1	6	5	6	7	2	5	3	4	1	2	4	7

Exercice 4 (Création de processus, fichiers et tubes (8 pts))
1) Soit le programme source suivant :

```
#include <stdio.h>
extern int e =3;
int main(int argc,char* argv[]){
  int p = 3;
  int t = 7;
  char *envp[] = { NULL };
  char *arg[] = { "/bin/ps", NULL };
  t = fork();
  if (t){
    p = 4;
    e = 5 ;
    printf("%d \t %d \t %d \n",e,p,t);
  }
  printf("%d \t %d \t %d \n",e,p,t);
  execve("/bin/ps", arg, envp);
  printf("%d \t %d \t %d \n",e,p,t);
  exit(0);
}
```

Donner l'affichage réalisé par ce programme.

2) Ecrire le code d'un processus qui créé un processus fils qui créera lui même un troisième processus qui écrira son identifiant dans un fichier "p3.txt" puis se terminera. Les 2 autres processus attendront alors la fin de l'exécution de leur fils respectifs pour écrire leur identifiant dans un fichier ("p2.txt" et "p1.txt") puis se termineront. **Vous devrez impérativement utiliser les appels système unix relatifs aux fichiers**.

3) Écrire un programme permettant de transférer le contenu d'un fichier texte passé en parametre d'un processus père vers un processus fils en utilisant un tube. Le fils affichera le contenu du fichier reçu.

Annexes

- `void (*signal(int sig, void (*func)(int)))(int);`
- `int beep(void);`
- `int close(int fd);`
- `int execlp(const char *file, const char *arg, ... /*, (char *)0 */);`
- `int flash(void);`
- `int fork();`
- `int kill(pid_t pid, int signo);`
- `int open(const char *pathname, int flags, mode_t mode);`
- `int pipe(int filedes[2]);`
- `int printf(const char * restrict format, ...);`
- `int rand(void);`
- `ssize_t read(int fd, void *buf, size_t count);`
- `int scanf(const char * restrict format, ...);`
- `void (*signal(int sig, void (*func)(int)))(int);`
- `void srand(unsigned seed);`
- `int system(const char *string);`
- `int raise(int signo);`
- `ssize_t write(int fd, const void *buf, size_t count);`
- `int wait(int * status);`