

1. A l'aide d'un schema expliquer la notion d'adressage virtuel et sa relation avec la mémoire physique.
2. Que fait le programme suivant ?
3. A quoi sert l'entrée "install" du makefile ?
4. Décrivez le code et son fonctionnement en détail.
5. Qu'est q'un défaut de page mineur / majeur ?
6. Le programme effectue-t-il plus de défaut de page lorsque le paramètre "lock" est passé ou sans paramètre ?
7. Dans un cas général pourquoi utiliser la fonction mlock ?

memory.c

```
1  #include <errno.h>
2  #include <sys/time.h>
3  #include <sys/resource.h>
4  #include <sys/mman.h>
5  #include <unistd.h>
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <string.h>
9
10 #define DATA "DonneesPriveesSensibles"
11 #define GIGA 1073741824L //nombre d'octets dans 1 Go
12 #define MAXDATASIZE (7*GIGA) //Taille de la memoire de la machine ou est
    exectue ce programme
13
14 void exit_err(const char* str)
15 {
16     perror(str);
17     exit(EXIT_FAILURE);
18 }
19
20 int main(int argc, char* argv[])
21 {
22
23     struct rlimit myLimits;
24     struct rusage myUsage;
25     char *ptr;
26     long int i;
27     int lock = 0;
28
29     if((argc > 1) && (strcmp("lock", argv[1]) == 0))
30         lock = 1;
31
32     myLimits.rlim_cur = RLIM_INFINITY;
33     myLimits.rlim_max = RLIM_INFINITY;
34     if(setrlimit(RLIMIT_MEMLOCK, &myLimits) == -1)
35         exit_err("Impossible de regler la limite RLIMIT_MEMLOCK");
36 }
```

```

37
38     if(setuid(getuid()) == -1)
39         exit_err("Impossible de regler l'uid");
40
41
42     if(lock) {
43         if(mlockall(MCL_FUTURE) == -1)
44             exit_err("Impossible de bloquer la memoire");
45         printf("Memory locked\n");
46     }
47
48     if((ptr = malloc(MAXDATASIZE)) == NULL)
49         exit_err("Impossible d'allouer de la memoire");
50
51     //Initialisation des donnees
52     for(i=0; i < MAXDATASIZE; i++)
53         *(ptr+i) = 0;
54
55     //Placement des donnees en memoire
56     for(i=0; i < MAXDATASIZE; i+=sizeof(DATA))
57         strcpy(ptr+i, DATA);
58
59     if(getrusage(RUSAGE_SELF, &myUsage) == -1)
60         exit_err("Impossible de lire les usages du processus");
61
62     printf("Number de défauts de pages majeurs: %ld\n", myUsage.
        ru_majflt);
63
64     if(lock)
65         if(munlockall() == -1)
66             exit_err("Impossible de debloquer la memoire");
67
68     return 0;
69 }

```

makefile

```

1 memory: memory.c
2     gcc memory.c -Wall -g -o memory
3
4 install:
5     @echo "L'installation necessite les droits root"
6     chown root ./memory
7     chmod u+s ./memory
8     @echo "Installation reussie"

```