

1. Que fait le programme suivant ?
2. Quel est l'impact de ce programme sur les ressources de l'ordinateur, notamment en matière de mémoire vive ?
3. A la ligne 16, est-ce que le code d'erreur `errno` est nécessairement le même que le code d'erreur rapporté par `perror` ?
4. A la ligne 22, à quoi correspond la valeur de `WEXITSTATUS(status)` pas rapport au processus enfant (i.e. à quelle autre variable est-elle reliée) ?
5. Ce programme peut-il créer des zombies ? Pourquoi ?

#### forkbomb.c

```
1 #include <sys/types.h>
2 #include <sys/wait.h>
3 #include <unistd.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <errno.h>
7
8 int main(int argc, char* argv []) {
9     pid_t pid = 0;
10    int status;
11
12    while(pid == 0) {
13        pid = fork();
14        if (pid < 0) {
15            perror("fork");
16            exit(errno);
17        }
18    }
19
20    waitpid(pid, &status, 0);
21    if WIFEXITED(status) {
22        printf("status: %d\n", WEXITSTATUS(status));
23    }
24
25    return errno;
26 }
```

#### makefile

```
1 all: forkbomb
2
3 forkbomb: forkbomb.c
4     gcc forkbomb.c -o forkbomb
```