

Exercise

Given the following program:

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

int main(int argc, char *argv[] )
{
    pid_t pid;
    int status;

    pid = fork();
    if (pid != 0)
    {
        while (pid != wait(&status));
    }
    else
    {
        sleep(5);
        exit(5);
    }

    pid = fork();
    if (pid != 0)
    {
        while (pid != wait(&status));
    }
    else
    {
        sleep(1);
        exit(1);
    }
}
```

In this program a new process is created and the program waits for the process termination. After that, a new process is created and the whole procedure is repeated.

Modify this code to create two different processes (that are executed in parallel) and the parent process waits for the termination of both of them.



Solution

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

int main(int argc, char *argv[] )
{
    pid_t pid;
    int status;

    pid = fork();
    if (pid == 0)
    {
        sleep(5);
        exit(5);
    }

    pid = fork();
    if (pid == 0)
    {
        sleep(1);
        exit(1);
    }

    wait(&status);
    wait(&status);
}
```