```c
/* Forcing sychronisation between parent and child
 */
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    // point of choosing which process to execute
    int pid = fork();

    if (pid == 0)
    {
        printf("I am the child: %d\n", getpid());
        int i;
        for (i = 0; i < 100; ++i)
            printf("child: %d\n", i);
        printf("\n");

        return 0;
    } else if (pid > 0) {
        printf("I am the parent: %d\n", getpid());

        int status;
        int result = wait(&status);
        if (-1 == result) {
            // failed case
        }

        int i;
        for (i = 0; i < 100; ++i)
            printf("parent: %d\n", i);
        printf("\n");

    } else {
        // fail case
    }

    return 0;
}
```