



COS40003 Concurrent Programming

Lecture 4.0: Process FAQ

The exercise in Lecture 2

```
#include <stdio.h>
#include <unistd.h>
int main(){
    printf("Hello World\n");
    fork();
    fork();
    fork();
    printf("Hello World\n");
    return 0;
}
```

Online C compilers

- Google: online c compiler
 - Here are three of them
- https://www.onlinegdb.com/online_c_compiler
- <https://repl.it/languages/c>
- https://rextester.com/l/c_online_compiler_gcc

Problem 1: printf() function

- Change the first printf by removing “\n”, see what happens
- `printf("a string");` -> put “a string” into buffer
- `printf("a string\n");` -> print “a string” onto screen immediately
- If not working, use
 - `fflush(stdout);` // Will now print everything in the stdout buffer

Problem 2: Parent process finished

- Problem: printing a random number of “hello world”.
- Reason: if the parent process finishes, the child processes stop
- To check how many processes:
 - add `sleep(1);`

Modified code Lecture 2

```
#include <stdio.h>
#include <unistd.h>
int main(){
    printf("Hello World\n\n");
    // or fflush(stdout);
    fork();
    fork();
    fork();
    printf("Hello World\n");
    sleep(1);
    return 0;
}
```