

Practical Work Series 1

Exercise 1: Compiling C programs with GCC

Compile the following C program using GCC and then answer the questions that follow:

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

int main() {
    char *table[] = {"C", "O", "M", "P", "U", "T", "E", "R", "S", "C", "I", "E", "N",
"C", "E"};
    int taille = sizeof(table) / sizeof(table[0]);

    for (int i = 0; i < taille; i++) {
        printf("%s\n", table[i]);
        sleep(1);
    }

    return 0;
}
```

- Indicate the process ID (PID) of this process.
- Run the process in background.
- Put the process to the foreground.
- Pause the execution of this process.
- Resume the execution of the suspended process.
- Stop the execution of this process.

Exercise 2:

For the following three programs, answer the following questions:

- Predict what will be displayed on the screen during execution.
- How many processes are created during execution?
- Provide the process hierarchy.

- **Program 1:**

```
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>
int main()
{
    int id, i ;
    fork();
    printf("A\n");
    if ((id=fork()==0)) {
        printf("B\n");
        exit(0);
    }
    wait(0);
    printf("C\n");
    return 0;
}
```

- **Program 2:**

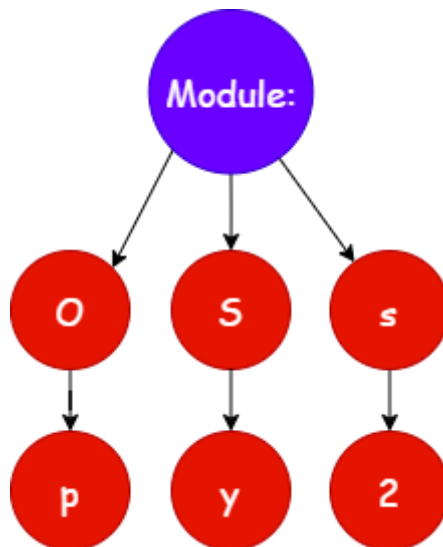
```
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>
void main()
{
    int id, i ;
    for (i=0 ; i<2; i++){
        if (fork()==0) {
            printf("A\n");
            exit(0);
        }
    }
    wait(0);
    printf("B\n");
    return 0;
}
```

- **Program 3:**

```
#include<sys/types.h>
#include<unistd.h>
#include<stdio.h>
#include <stdlib.h>
int main(){
    printf("Good morning ");
    if(fork () == 0)
    {
        printf("Sir\n");
        exit(0);
    }
    else
    {
        printf("Mrs\n");
        wait(NULL);
        exit(0);
    }
    return 0 ;
}
```

Exercise 3:

Write a C program that creates the following tree structure, and then, using the *wait(0)* and *exit(0)* functions, synchronize between the different processes to display:
Module: OpSys.



Exercise 4 (Optional):

Write a C program that creates the following tree structure, and then, using the functions *wait(0)* et *exit(0)*, synchronize between the different processes to display: Student: B3SSE.

