

## Assignment on Process

1. Write a C Program that redirects standard output to a file. Ex: ls > file
2. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
3. Write a C program to create a Zombie Process.
4. Write a C Program to create an Orphan process.
5. How do fork () and exec () functions differ? How many new processes will be created if the following sequence is executed: fork (); fork (); fork ();
6. What are the three features that are provided by waitpid() but not by wait()?
7. What is a zombie process? Write a program that will avoid zombie processes.
8. Analyze the behavior of the following Program

```
int main ()
{
    int a = 10, b = 20, c;
    if (!fork ())
    {
        a = b + 10;
        c = a + b + 2;
        printf ("1st case: C=%d\n", c);
        exit (0);
    }
    else
    {
        b = a + 10;
        c = a + b + 10;
        printf ("2nd Case: C=%d", c);
    }
    return 1;
}
```

9. Write a C Program for creating a new process and run “ls -l” in the newly created process.
10. Implement the above program using different exec family functions.
11. Write a program that executes the "cat -b -v -t filename" command. Call your executable myfork

Details:

- The call to your program will be made with the following command:  
% myfork filename
- Your code will fork()
- The child will use the execl to call cat and use the filename passed as an argument on the command line
- The parent will wait for the child to finish
- Your program will also print from the child process:
  - The process id
  - The parent id
  - The process group id

- and print from the parent process:
  - the process id
  - the parent id
  - the process group id
- Comment out the `execl` call and add instead a call to `execv`. Add any necessary variables to do that.

12. Answer the following questions:

- If you try to print a message after the `exec*` call, does it print it? Why? Why not?
- Who is the parent of your executable (myfork) program?