```c

/*Author:   Jesus Minjares
 *Lab 01
    Write a program that creates three processes to execute the
    following three programs with command line arguments:
    1. “cat /proc/cpuinfo”
    2. “echo Hello World”
    3. “uname -a”
    and then waits for these three processes to complete before creating a process to execute
    “ls -l“ and immediately printing “Goodbye!” and exiting; not waiting for the last process to
    complete.
    -----------------------------------------
    excelp(command, command, arguments, NULL)
 */
#include <sys/wait.h>
#include <sys/types.h> //fork()
#include <unistd.h> //execl()
#include <stdlib.h>
#include <stdio.h>
int main(){
    pid_t child[4]; //create 4 pid_t for each child of the parent
    char *commands[] =  {"cat","echo","uname", "ls"}; //commands
    char *arguments[]= {"/proc/cpuinfo","Hello World","-a", "-l"}; //arguments
    for(int i =0; i < 3; i++){ //iterate over the first 3 childs
        child[i] = fork(); //fork a child
        if(child[i] < 0){ //if child at i is negative, exit program
            fprintf(stderr,"Error forking a process\n"); //error message
            return -1; //exit
        }
        else if(child[i] == 0)//if the child at i is 0, then execute program
            execlp(commands[i], commands[i], arguments[i], NULL);
        else
            waitpid(child[i] ,NULL, 0); //as the parent wait for the childs to end
    }
    //after the 3 child are done
    child[3] = fork(); //fork another child
    if(child[3] < 0){ //if negative, an error has occured
        fprintf(stderr,"Error forking a process\n");//error message
        return -1; //exit
    }
    else if( child[3] == 0) //if the last child pid is 0, then execute program
        execlp(commands[3], commands[3], arguments[3], NULL);
    else
        printf("Good bye!!!\n"); //dont wait for the last process to end
    return 0;
}
```