|  |
| --- |
| #include <iostream>  #include <unistd.h>  using namespace std;    // Driver Code  int main()  {      int pid;      pid = fork();      if (pid == 0)      {          cout << "\nParent Process id : "               << getpid() << endl;          cout << "\nChild Process with parent id : "               << getppid() << endl;      }      return 0;  } |
|  |
| #include <stdio.h>  #include <sys/types.h>  int main()  {      fork();      fork();      fork();      printf("hello\n");      return 0;  } |
| hello  hello  hello  hello  hello  hello  hello  hello |
| #include <stdio.h>  #include <sys/types.h>  #include <unistd.h>    void forkexample()  {      int x = 1;        if (fork() == 0)          printf("Child has x = %d\n", ++x);      else          printf("Parent has x = %d\n", --x);  }  int main()  {      forkexample();      return 0;  } |
| Parent has x = 0  Child has x = 2  (or)  Child has x = 2  Parent has x = 0 |
| #include <stdio.h>  #include <unistd.h>  int main()  {      fork();      fork() && fork() || fork();      fork();        printf("forked\n");      return 0;  } |
| forked  forked  forked  forked  forked  forked  forked  forked  forked  forked  forked |
| // C++ program to demonstrate searching  // in parent and printing result  // in child processes using fork()  #include <iostream>  #include <unistd.h>  using namespace std;    // Driver code  int main()  {        int key = 10;      int id = fork();        // Checking value of process id returned by fork      if (id > 0)  {            cout << "Parent process \n";            int a[] = { 3, 8, 4, 10, 80 };          int n = 5;          int flag;          int i;            for (i = 0; i < n; i++)          {                if (a[i] != key) {                  flag = 0;              }                else  {                    flag = 1;              }          }            if (flag == 1) {                cout << "key is not present in array";          }            else {                cout << "key is present in array";              cout << "\n";            }      }        // If n is 0 i.e. we are in child process      else {            cout << "Child process \n";          cout << "numbers to be search is ";          cout << key;      }        return 0;  } |
| <https://www.geeksforgeeks.org/multiple-calculations-4-processes-using-fork/> |