

**Міністерство освіти і науки України**  
**Національний технічний університет України**  
**"Київський політехнічний інститут імені Ігоря Сікорського"**  
**Фізико-технічний інститут**

**ОПЕРАЦІЙНІ СИСТЕМИ**

Комп'ютерний практикум

Робота № 6

Варіант 8

**Виконав:**

Студент групи ФБ-95

Прохоренко Я.Ю.

## Створення процесів у Linux із застосуванням системних викликів `fork()` і `exec()`

Мета: Оволодіння практичними навичками застосування системних викликів у програмах, дослідження механізму створення процесів у UNIX-подібних системах.

Operating System: Ubuntu 20.10

Kernel: Linux 5.8.0-43-generic

Architecture: x86-64

Дата виконання: 18.02.2021

```
augfitzr@ubuntu:~$ g++ -o myforktest myforktest.cpp
augfitzr@ubuntu:~$ ./myforktest
Parent Process: Global variable: 2 Stack variable: 20
Child Process: Global variable: 3 Stack variable: 21
```

Спочатку батьківський потім дочірній, цей порядок дотримується завжди.

```
3     }
4     else // parent
5     {
6         // Code only executed by parent process
7         sleep(55);
8         sIdentifier = "Parent Process:";
9     }
10    }
```

```
augfitzr@ubuntu:~$ g++ -o myforktest myforktest.cpp
augfitzr@ubuntu:~$ ./myforktest
Child Process: Global variable: 3 Stack variable: 21
Parent Process: Global variable: 2 Stack variable: 20
```

Результати виконання не змінилися.

```
pid_t pID = fork();
for(int i = 0; i < 10; i++){
    if (pID == 0) // child
    {
```

```

augfitzr@ubuntu:~$ g++ -o myforktest myforktest.cpp
augfitzr@ubuntu:~$ ./myforktest
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Parent Process: Global variable: 2 Stack variable: 20
Child Process: Global variable: 3 Stack variable: 21
Child Process: Global variable: 4 Stack variable: 22
Child Process: Global variable: 5 Stack variable: 23
Child Process: Global variable: 6 Stack variable: 24
Child Process: Global variable: 7 Stack variable: 25
Child Process: Global variable: 8 Stack variable: 26
Child Process: Global variable: 9 Stack variable: 27
Child Process: Global variable: 10 Stack variable: 28
Child Process: Global variable: 11 Stack variable: 29
Child Process: Global variable: 12 Stack variable: 30

```

В батьківському процесі значення залишається не змінним, а значення дочірнього процесу інкрементується, оскільки fork копіює адресний простір для нової програми.

```

augfitzr@ubuntu:~$ g++ -o myforktest myforktest.cpp
augfitzr@ubuntu:~$ ./myforktest
Child Process: Global variable: 3 Stack variable: 21
Parent Process: Global variable: 3 Stack variable: 21
*** stack smashing detected ***: terminated
Aborted (core dumped)

```

Vfork не копіює адресний простір для нового процесу.

Виникає помилка stack smashing, оскільки дочірній процес не має прав модифікувати значення в стеку.

Для коректної роботи слід додати `_exit()`; і дані будуть модифікуватись у батьківському процесі.



```

if (pID == 0) // child
{
    // Code only executed by child process

    sIdentifier = "Child Process: ";
    globalVariable++;
    iStackVariable++;
    execl("/bin/ls", "/bin/ls", "-l", "-a", (char *) 0);
}

```

```

augfitzr@ubuntu:~$ g++ -o myforktest myforktest.cpp
augfitzr@ubuntu:~$ ./myforktest
Parent Process: Global variable: 3 Stack variable: 21
augfitzr@ubuntu:~$ total 20444
drwxr-xr-x 17 augfitzr augfitzr      4096 Feb 18 15:26 .
drwxr-xr-x  3 root      root          4096 Feb 18 11:05 ..
-rw-r--r--  1 augfitzr augfitzr       279 Feb 18 14:26 .bash_history
-rw-r--r--  1 augfitzr augfitzr       220 Feb 18 11:05 .bash_logout
-rw-r--r--  1 augfitzr augfitzr      3771 Feb 18 11:05 .bashrc
drwxr-xr-x 15 augfitzr augfitzr      4096 Feb 18 14:32 .cache
drwxr-xr-x 16 augfitzr augfitzr      4096 Feb 18 15:11 .config
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:10 Desktop
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Documents
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 15:09 Downloads
-rw-rw-r--  1 augfitzr augfitzr 20785984 Feb 18 14:43 file
drwxr-xr-x  3 augfitzr augfitzr      4096 Feb 18 14:27 .gnupg
drwxr-xr-x  3 augfitzr augfitzr      4096 Feb 18 14:08 .local
-rw-rw-r--  1 augfitzr augfitzr    30520 Feb 18 14:37 log
drwxr-xr-x  5 augfitzr augfitzr      4096 Feb 18 14:09 .mozilla
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Music
-rwxrwxr-x  1 augfitzr augfitzr    18448 Feb 18 15:26 myforktest
-rw-rw-r--  1 augfitzr augfitzr       973 Feb 18 15:25 myforktest.cpp
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 15:26 Pictures
-rw-r--r--  1 augfitzr augfitzr       807 Feb 18 11:05 .profile
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Public
-rw-rw-r--  1 augfitzr augfitzr        66 Feb 18 14:53 .selected_editor
drwxrwxr-x  3 augfitzr augfitzr      4096 Feb 18 14:31 .shutter
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 .ssh
-rw-r--r--  1 augfitzr augfitzr         0 Feb 18 14:24 .sudo_as_admin_s
successful
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Templates
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Videos

```

```

if (pID == 0) // child
{
    // Code only executed by child process

    sIdentifier = "Child Process: ";
    globalVariable++;
    iStackVariable++;

    execl("/bin/ps", "/bin/ps", "-a", "-u", "-x", (char *)0);
}
else if (pID < 0) // failed to fork
{
    cerr << "Failed to fork" << endl;
    exit(1);
    // Throw exception
}
else // parent
{
    // Code only executed by parent process
    //sleep(55);
    sIdentifier = "Parent Process: ";
    execl("/bin/ls", "/bin/ls", "-l", "-a", (char *) 0);
}

```

```

augfitzr@ubuntu:~$ g++ -o myforktest myforktest.cpp
augfitzr@ubuntu:~$ ./myforktest
total 20444
drwxr-xr-x 17 augfitzr augfitzr      4096 Feb 18 15:29 .
drwxr-xr-x  3 root     root         4096 Feb 18 11:05 ..
-rw-r--r--  1 augfitzr augfitzr       279 Feb 18 14:26 .bash_history
-rw-r--r--  1 augfitzr augfitzr       220 Feb 18 11:05 .bash_logout
-rw-r--r--  1 augfitzr augfitzr      3771 Feb 18 11:05 .bashrc
drwxr-xr-x 15 augfitzr augfitzr      4096 Feb 18 14:32 .cache
drwxr-xr-x 16 augfitzr augfitzr      4096 Feb 18 15:11 .config
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:10 Desktop
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Documents
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 15:09 Downloads
-rw-rw-r--  1 augfitzr augfitzr 20785984 Feb 18 14:43 file
drwxr-xr-x  3 augfitzr augfitzr      4096 Feb 18 14:27 .gnupg
drwxr-xr-x  3 augfitzr augfitzr      4096 Feb 18 14:08 .local
-rw-rw-r--  1 augfitzr augfitzr     30520 Feb 18 14:37 log
drwxr-xr-x  5 augfitzr augfitzr      4096 Feb 18 14:09 .mozilla
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Music
-rwxrwxr-x  1 augfitzr augfitzr     18448 Feb 18 15:29 myforktest
-rw-rw-r--  1 augfitzr augfitzr     1037 Feb 18 15:29 myforktest.cpp
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 15:29 Pictures
-rw-r--r--  1 augfitzr augfitzr       807 Feb 18 11:05 .profile
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Public
-rw-rw-r--  1 augfitzr augfitzr        66 Feb 18 14:53 .selected_editor
drwxrwxr-x  3 augfitzr augfitzr      4096 Feb 18 14:31 .shutter
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 .ssh
-rw-r--r--  1 augfitzr augfitzr         0 Feb 18 14:24 .sudo_as_admin_successful
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Templates
drwxr-xr-x  2 augfitzr augfitzr      4096 Feb 18 14:09 Videos
augfitzr@ubuntu:~$ USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.1  0.3 104480 12908 ?        Ss   14:26   0:07 /sb
root         2  0.0  0.0     0     0 ?        S    14:26   0:00 [kt
root         3  0.0  0.0     0     0 ?        I<   14:26   0:00 [rc
root         4  0.0  0.0     0     0 ?        I<   14:26   0:00 [rc
root         6  0.0  0.0     0     0 ?        I<   14:26   0:00 [kw
root         9  0.0  0.0     0     0 ?        I<   14:26   0:00 [mm
root        10  0.0  0.0     0     0 ?        S    14:26   0:00 [ks

```

Waitpid()

Getpid()

Getppid()

**Висновки:** за допомогою спеціальних команд можна дуже зручно створювати батьківські та дочірні процеси, і зручно керувати їми виконуючі певні дії.