Sebastian Ratańczuk rs44476

Zadanie 1

Kod programu:

home > user > Pulpit > C lab1.c > main()

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

int main(){

    pid_t pid = fork();

    if(pid<0){
        printf("Error\n");
        return -1;
    }

    if(pid==0){
        printf("I'm parent. My pid is: %d\n", getpid());
    }
    else{
        printf("I'm child My pid is: %d\n", getpid());
    }

    return 0;
}
```

```
user@user-VirtualBox:~/Pulpit$ ./a.out
I'm child My pid is: 7940
user@user-VirtualBox:~/Pulpit$ I'm parent. My pid is: 7941
```

Sebastian Ratańczuk rs44476

```
user@user-VirtualBox:~/Pulpit$ strace ./a.out
execve("./a.out", ["./a.out"], 0x7ffe9a8618e0 /* 49 vars */) = 0
brk(NULL)                               = 0x55c2f92be000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd5e32bf40) = -1 EINVAL (Zły argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Nie ma takiego pliku ani katalogu)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=66989, ...}) = 0
mmap(NULL, 66989, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f2362897000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f2362895000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f23626a3000
mprotect(0x7f23626c8000, 1847296, PROT_NONE) = 0
mmap(0x7f23626c8000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f23626c8000
mmap(0x7f2362840000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f2362840000
mmap(0x7f236288b000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f236288b000
mmap(0x7f2362891000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2362891000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f2362896540) = 0
mprotect(0x7f236288b000, 12288, PROT_READ) = 0
mprotect(0x55c2f7a6b000, 4096, PROT_READ) = 0
mprotect(0x7f23628d5000, 4096, PROT_READ) = 0
munmap(0x7f2362897000, 66989)           = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7f2362896810) = 7955
getpid()                                = 7954
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL)                               = 0x55c2f92be000
brk(0x55c2f92df000)                     = 0x55c2f92df000
write(1, "I'm child My pid is: 7954\n", 26I'm child My pid is: 7954
) = 26
exit_group(0)                           = ?
+++ exited with 0 +++
user@user-VirtualBox:~/Pulpit$ I'm parent. My pid is: 7955
```

```
user@user-VirtualBox:~/Pulpit$ strace -f ./a.out
execve("./a.out", ["./a.out"], 0x7ffd10c204f8 /* 49 vars */) = 0
brk(NULL)                               = 0x558dc8c12000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcf2e532c0) = -1 EINVAL (Zły argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Nie ma takiego pliku ani katalogu)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=66989, ...}) = 0
mmap(NULL, 66989, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f54669b6000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f54669b4000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f54667c2000
mprotect(0x7f54667e7000, 1847296, PROT_NONE) = 0
mmap(0x7f54667e7000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f54667e7000
mmap(0x7f546695f000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f546695f000
mmap(0x7f54669aa000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f54669aa000
mmap(0x7f54669b0000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f54669b0000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f54669b5540) = 0
mprotect(0x7f54669aa000, 12288, PROT_READ) = 0
mprotect(0x558dc790f000, 4096, PROT_READ) = 0
mprotect(0x7f54669f4000, 4096, PROT_READ) = 0
munmap(0x7f54669b6000, 66989)           = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7f54669b5810) = 2689
getpid()                                = 2688
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
brk(NULL)                               = 0x558dc8c12000
brk(0x558dc8c33000)                     = 0x558dc8c33000
write(1, "I'm child. My pid is: 2688\n", 27I'm child. My pid is: 2688
) = 27
exit_group(0)                           = ?
+++ exited with 0 +++
strace: Process 2689 attached
getpid()                                = 2689
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
brk(NULL)                               = 0x558dc8c12000
brk(0x558dc8c33000)                     = 0x558dc8c33000
write(1, "I'm parent. My pid is: 2689\n", 28I'm parent. My pid is: 2689
) = 28
exit_group(0)                           = ?
+++ exited with 0 +++
user@user-VirtualBox:~/Pulpit$
```

execve() – odpowiada za uruchomienie programu w nowym procesie
clone() – odpowiada wywołaniu funkcji fork(), czyli za zduplikowanie procesu (linia 7)

```
7    pid_t pid = fork();
```

getpid() – wywołanie funkcji getpid(), która zwraca nam numer PID (linia 15 i 18)
write() – odpowiada wywołaniu printf() (linia 15 i 18)

Sebastian Ratańczuk rs44476

```
14      if(pid==0){
15          printf("I'm parent. My pid is: %d\n", getpid());
16      }
17      else{
18          printf("I'm child My pid is: %d\n", getpid());
19      }
```

Zadanie 2

```
user@user-VirtualBox:~/Pulpit$ gcc -g lab1.c
user@user-VirtualBox:~/Pulpit$ ./a.out
^C
user@user-VirtualBox:~/Pulpit$ ./a.out&
[1] 8285
```

```
user@user-VirtualBox:~/Pulpit$ sudo gdb
[sudo] hasło użytkownika user:
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) attach 8297
Attaching to process 8297
Reading symbols from /home/user/Pulpit/a.out...
Reading symbols from /lib/x86_64-linux-gnu/libc.so.6...
Reading symbols from /usr/lib/debug//lib/x86_64-linux-gnu/libc-2.31.so...
Reading symbols from /lib64/ld-linux-x86-64.so.2...
(No debugging symbols found in /lib64/ld-linux-x86-64.so.2)
main () at lab1.c:8
8           while(a);
(gdb) print a
$1 = 1
(gdb) set variable a=0
(gdb) print a
$2 = 0
(gdb) c
Continuing.
[Inferior 1 (process 8297) exited normally]
(gdb)
```

Wstępnie skompilowałem program z przełącznikiem -g. Uruchomiłem program w tle wykorzystując ./a.out& otrzymałem w ten sposób pid procesu. Następnie uruchomiłem gdb z uprawnieniami roota i podpiąłem się do procesu wykorzystując attach pid. Wypisałem zmienna a oraz zmieniłem ją na 0, następnie kontynuowałem wykonywanie programu poleceniem c, przez co program się zakończył.

Sebastian Ratańczuk rs44476

Zadanie 3

```
user@user-VirtualBox:~/Pulpit$ gcc lab1.c -g
```

```
user@user-VirtualBox:~/Pulpit$ valgrind --leak-check=yes ./a.out
==2164== Memcheck, a memory error detector
==2164== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2164== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==2164== Command: ./a.out
==2164==
==2164== Invalid write of size 1
==2164==    at 0x1091AF: main (lab1.c:10)
==2164==  Address 0x4a5004a is 0 bytes after a block of size 10 alloc'd
==2164==    at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2164==    by 0x10917E: main (lab1.c:6)
==2164==
==2164==
==2164== HEAP SUMMARY:
==2164==     in use at exit: 30 bytes in 2 blocks
==2164==   total heap usage: 2 allocs, 0 frees, 30 bytes allocated
==2164==
==2164== 10 bytes in 1 blocks are definitely lost in loss record 1 of 2
==2164==    at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2164==    by 0x10917E: main (lab1.c:6)
==2164==
==2164== 20 bytes in 1 blocks are definitely lost in loss record 2 of 2
==2164==    at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2164==    by 0x10918C: main (lab1.c:7)
==2164==
==2164== LEAK SUMMARY:
==2164==    definitely lost: 30 bytes in 2 blocks
==2164==    indirectly lost: 0 bytes in 0 blocks
==2164==      possibly lost: 0 bytes in 0 blocks
==2164==    still reachable: 0 bytes in 0 blocks
==2164==         suppressed: 0 bytes in 0 blocks
==2164==
==2164== For lists of detected and suppressed errors, rerun with: -s
==2164== ERROR SUMMARY: 3 errors from 3 contexts (suppressed: 0 from 0)
user@user-VirtualBox:~/Pulpit$
```

Raport programu Valgrind wykazał, że mamy 3 podatności w naszym programie

Dwie dotyczą braku zwalniania pamięci tablic typu unsigned char powołanych w (liniach 6 i 7), należałoby użyć polecenia free(), gdyby tablice były powoływane do życia w funkcji która jest często wykonywalna, nasza pamięć mogła by się zapełnić.

Trzeci błąd dotyczy próby zapisu do tablicy poza przydzieloną pamięcią (linia 10), nasza tablica zawiera 10 elementów a indeks 10 odnosi się do 11 elementu. Jeżeli chcemy odnieść się do ostatniego elementu należy użyć 9 indeksu.

Kod po poprawkach

```
lab1.c
File  Edit  Selection  View  Go  Run  Terminal  Help
C lab1.c
home > user > Pulpit > C lab1.c > main()
1  #include <stdlib.h>
2  #include <string.h>
3
4  int main() {
5
6      unsigned char *p1 = malloc(10*sizeof(unsigned char));
7      unsigned char *p2 = malloc(20*sizeof(unsigned char));
8
9      memset(p1, 1, 10);
10     p1[9] = 2;
11
12     free(p1);
13     free(p2);
14     return 0;
15 }
```

Sebastian Ratańczuk rs44476

Raport Valgrinda po poprawkach:

```
user@user-VirtualBox:~/Pulpit$ valgrind --leak-check=yes ./a.out
==2253== Memcheck, a memory error detector
==2253== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2253== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==2253== Command: ./a.out
==2253==
==2253==
==2253== HEAP SUMMARY:
==2253==     in use at exit: 0 bytes in 0 blocks
==2253==   total heap usage: 2 allocs, 2 frees, 30 bytes allocated
==2253==
==2253== All heap blocks were freed -- no leaks are possible
==2253==
==2253== For lists of detected and suppressed errors, rerun with: -s
==2253== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```