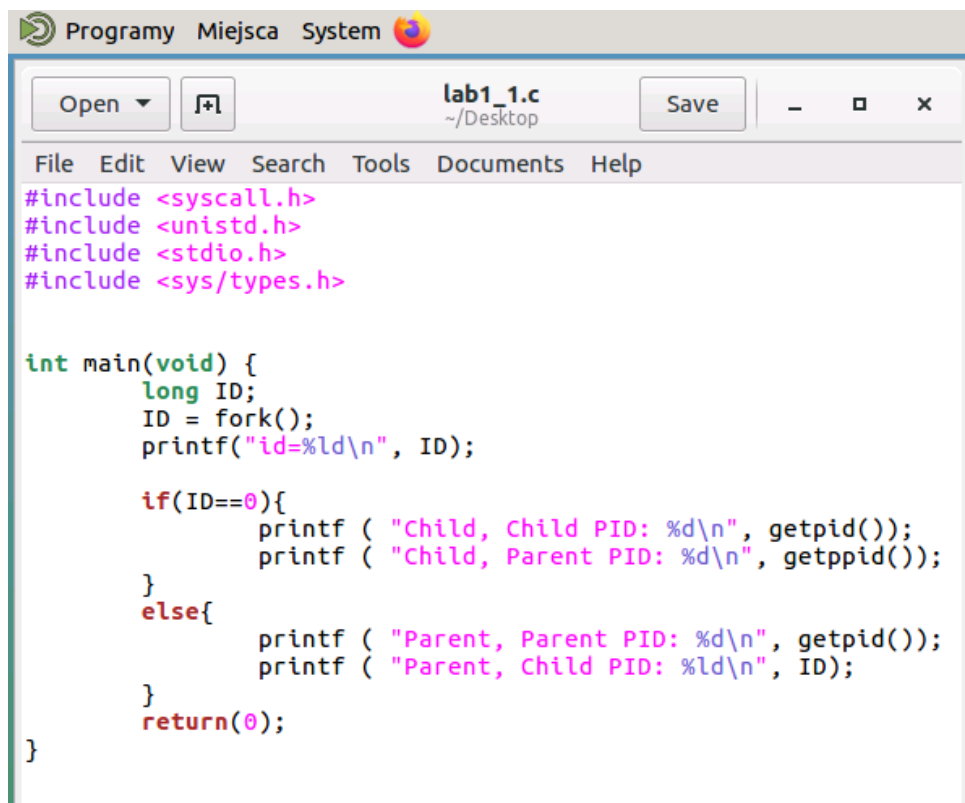


Kod programu:

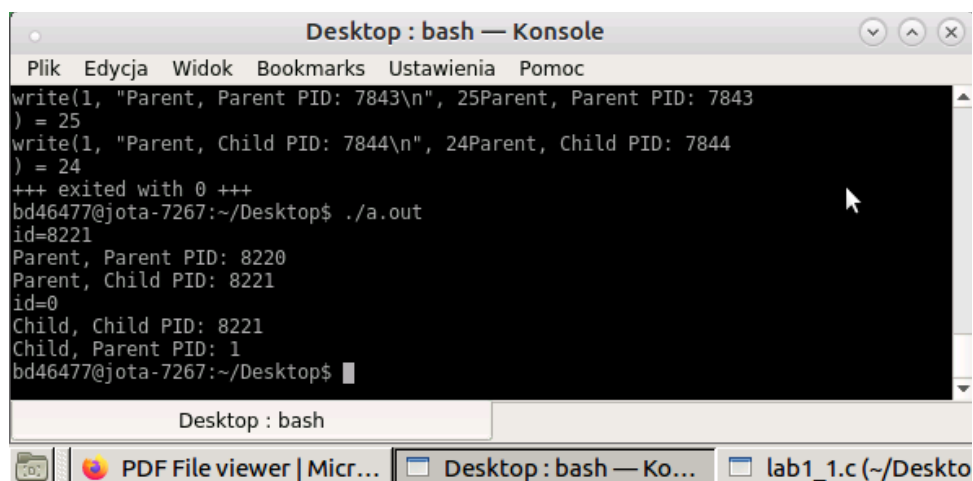


```
#include <syscall.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>

int main(void) {
    long ID;
    ID = fork();
    printf("id=%ld\n", ID);

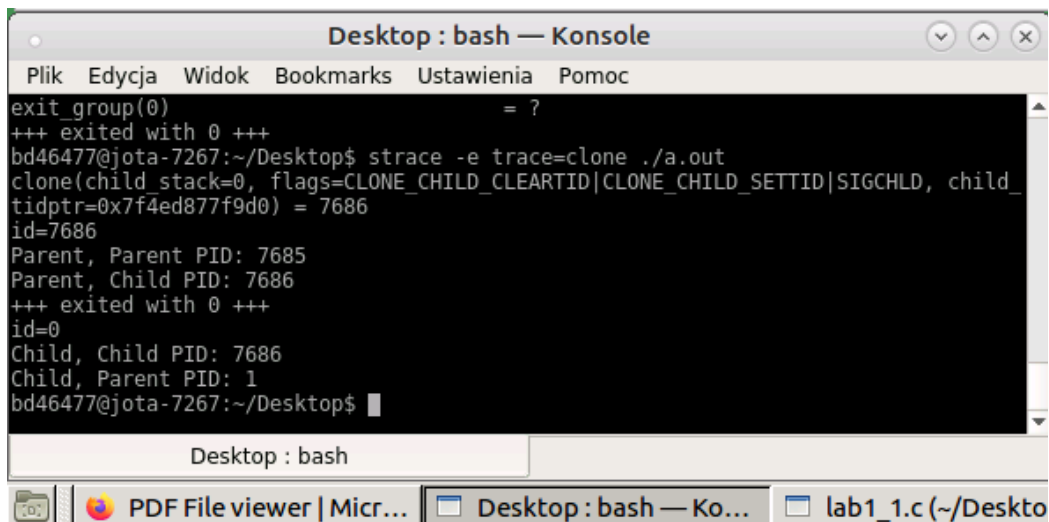
    if(ID==0){
        printf ( "Child, Child PID: %d\n", getpid());
        printf ( "Child, Parent PID: %d\n", getppid());
    }
    else{
        printf ( "Parent, Parent PID: %d\n", getpid());
        printf ( "Parent, Child PID: %ld\n", ID);
    }
    return(0);
}
```

Wywołanie programu:



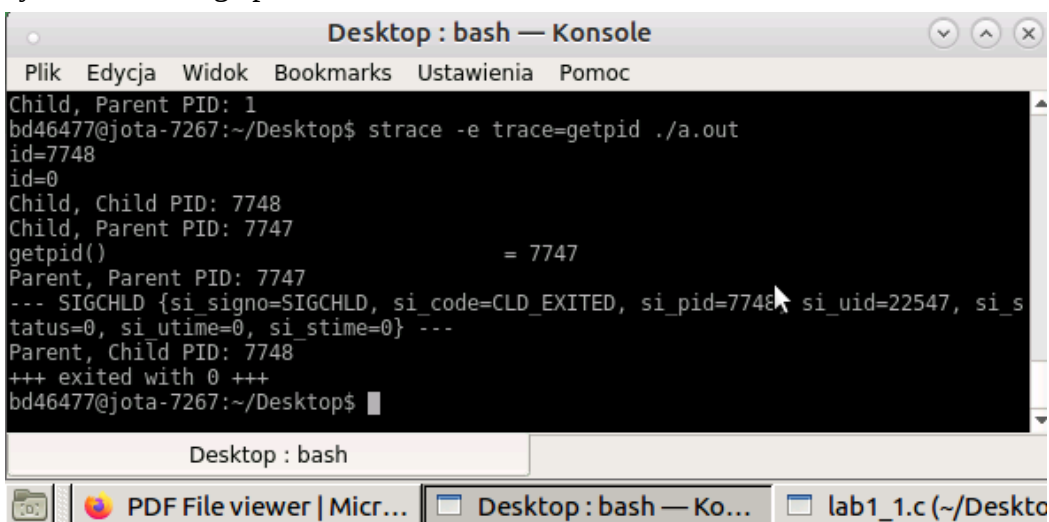
```
Desktop : bash — Konsole
Plik  Edycja  Widok  Bookmarks  Ustawienia  Pomoc
write(1, "Parent, Parent PID: 7843\n", 25Parent, Parent PID: 7843
) = 25
write(1, "Parent, Child PID: 7844\n", 24Parent, Child PID: 7844
) = 24
+++ exited with 0 +++
bd46477@jota-7267:~/Desktop$ ./a.out
id=8221
Parent, Parent PID: 8220
Parent, Child PID: 8221
id=0
Child, Child PID: 8221
Child, Parent PID: 1
bd46477@jota-7267:~/Desktop$
```

Wynik strace dla clone:



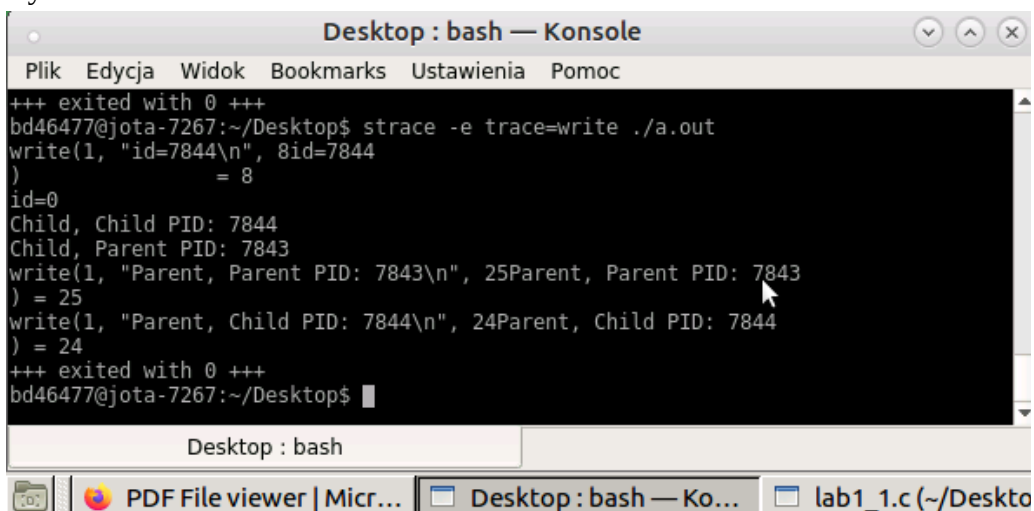
```
Desktop : bash — Konsole
Plik  Edycja  Widok  Bookmarks  Ustawienia  Pomoc
exit_group(0)                                = ?
+++ exited with 0 +++
bd46477@jota-7267:~/Desktop$ strace -e trace=clone ./a.out
clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_
tidptr=0x7f4ed877f9d0) = 7686
id=7686
Parent, Parent PID: 7685
Parent, Child PID: 7686
+++ exited with 0 +++
id=0
Child, Child PID: 7686
Child, Parent PID: 1
bd46477@jota-7267:~/Desktop$
```

Wynik strace dla getpid:



```
Desktop : bash — Konsole
Plik  Edycja  Widok  Bookmarks  Ustawienia  Pomoc
Child, Parent PID: 1
bd46477@jota-7267:~/Desktop$ strace -e trace=getpid ./a.out
id=7748
id=0
Child, Child PID: 7748
Child, Parent PID: 7747
getpid()                                = 7747
Parent, Parent PID: 7747
--- SIGCHLD {si signo=SIGCHLD, si code=CLD_EXITED, si pid=7748, si uid=22547, si_s
tatus=0, si utime=0, si stime=0} ---
Parent, Child PID: 7748
+++ exited with 0 +++
bd46477@jota-7267:~/Desktop$
```

Wynik strace dla write:



```
Desktop : bash — Konsole
Plik  Edycja  Widok  Bookmarks  Ustawienia  Pomoc
+++ exited with 0 +++
bd46477@jota-7267:~/Desktop$ strace -e trace=write ./a.out
write(1, "id=7844\n", 8id=7844
) = 8
id=0
Child, Child PID: 7844
Child, Parent PID: 7843
write(1, "Parent, Parent PID: 7843\n", 25Parent, Parent PID: 7843
) = 25
write(1, "Parent, Child PID: 7844\n", 24Parent, Child PID: 7844
) = 24
+++ exited with 0 +++
bd46477@jota-7267:~/Desktop$
```

```
Desktop : bash — Konsole
Plik  Edycja  Widok  Bookmarks  Ustawienia  Pomoc
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./a.out...done.
Attaching to program: /home/bd46477/Desktop/a.out, process 8614
Reading symbols from /lib/x86_64-linux-gnu/libc.so.6...Reading symbols from /usr/lib/debug//lib/x86_64-linux-gnu/libc-2.23.so...done.
done.
Reading symbols from /lib64/ld-linux-x86-64.so.2...Reading symbols from /usr/lib/debug//lib/x86_64-linux-gnu/ld-2.23.so...done.
done.
main () at lab1_2.c:9
9         while(a);
(gdb) l
4         #include <sys/types.h>
5
6
7         int main() {
8             int a=1;
9             while(a);
10            return 0;
11        }
(gdb) b 9
Breakpoint 1 at 0x4004e1: file lab1_2.c, line 9.
(gdb) c
Continuing.

Breakpoint 1, main () at lab1_2.c:9
9         while(a);
(gdb) p a
$1 = 1
(gdb) set var a=0
(gdb)
(gdb)
(gdb)
(gdb) q
A debugging session is active.

        Inferior 1 [process 8614] will be detached.

Quit anyway? (y or n) y
Detaching from program: /home/bd46477/Desktop/a.out, process 8614
[1]+  Zakończono ./a.out
bd46477@jota-7267:~/Desktop$
```

Do wykonania zadania musiałem skompilować z dodaniem -g, aby móc wejść do debuggera. Następnie po odpaleniu programu: ./a.out zatrzymuje proces aby wrzucić go w tło. Sprawdzam procesy za pomocą komendy ps. Kolejnym krokiem jest zmiana wartości a, wpisując gdb ./a.out id\_procesu, po uruchomieniu gdb wpisujemy set var a=0

```
Desktop : bash — Konsole
Plik Edycja Widok Bookmarks Ustawienia Pomoc
valgrind: Unknown option: --tool-memcheck
valgrind: Use --help for more information or consult the user manual.
bd46477@jota-7267:~/Desktop$ valgrind --tools-memcheck --leak-check=yes ./a.out
valgrind: Unknown option: --tools-memcheck
valgrind: Use --help for more information or consult the user manual.
bd46477@jota-7267:~/Desktop$ valgrind --tool-memcheck --leak-check=yes ./a.out
valgrind: Unknown option: --tool-memcheck
valgrind: Use --help for more information or consult the user manual.
bd46477@jota-7267:~/Desktop$ valgrind --leak-check=yes ./a.out
==9762== Memcheck, a memory error detector
==9762== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==9762== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==9762== Command: ./a.out
==9762==
==9762== Invalid write of size 1
==9762==    at 0x4005A8: main (lab1_3.c:8)
==9762==    Address 0x520404a is 0 bytes after a block of size 10 alloc'd
==9762==    at 0x4C2DB8F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-li
nux.so)
==9762==    by 0x400577: main (lab1_3.c:5)
==9762==
==9762== HEAP SUMMARY:
==9762==    in use at exit: 30 bytes in 2 blocks
==9762==    total heap usage: 2 allocs, 0 frees, 30 bytes allocated
==9762==
==9762== 10 bytes in 1 blocks are definitely lost in loss record 1 of 2
==9762==    at 0x4C2DB8F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-li
nux.so)
==9762==    by 0x400577: main (lab1_3.c:5)
==9762==
==9762== 20 bytes in 1 blocks are definitely lost in loss record 2 of 2
==9762==    at 0x4C2DB8F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-li
nux.so)
==9762==    by 0x400585: main (lab1_3.c:6)
==9762==
==9762== LEAK SUMMARY:
==9762==    definitely lost: 30 bytes in 2 blocks
==9762==    indirectly lost: 0 bytes in 0 blocks
==9762==    possibly lost: 0 bytes in 0 blocks
==9762==    still reachable: 0 bytes in 0 blocks
==9762==    suppressed: 0 bytes in 0 blocks
==9762==
==9762== For counts of detected and suppressed errors, rerun with: -v
==9762== ERROR SUMMARY: 3 errors from 3 contexts (suppressed: 0 from 0)
bd46477@jota-7267:~/Desktop$
```

Desktop : bash

03-04 10:56:14.png Page 3 of 3 64 words, 431 chara

PDF File viewer | Micr... Desktop : bash — Ko... lab1\_3.c (~/Desko

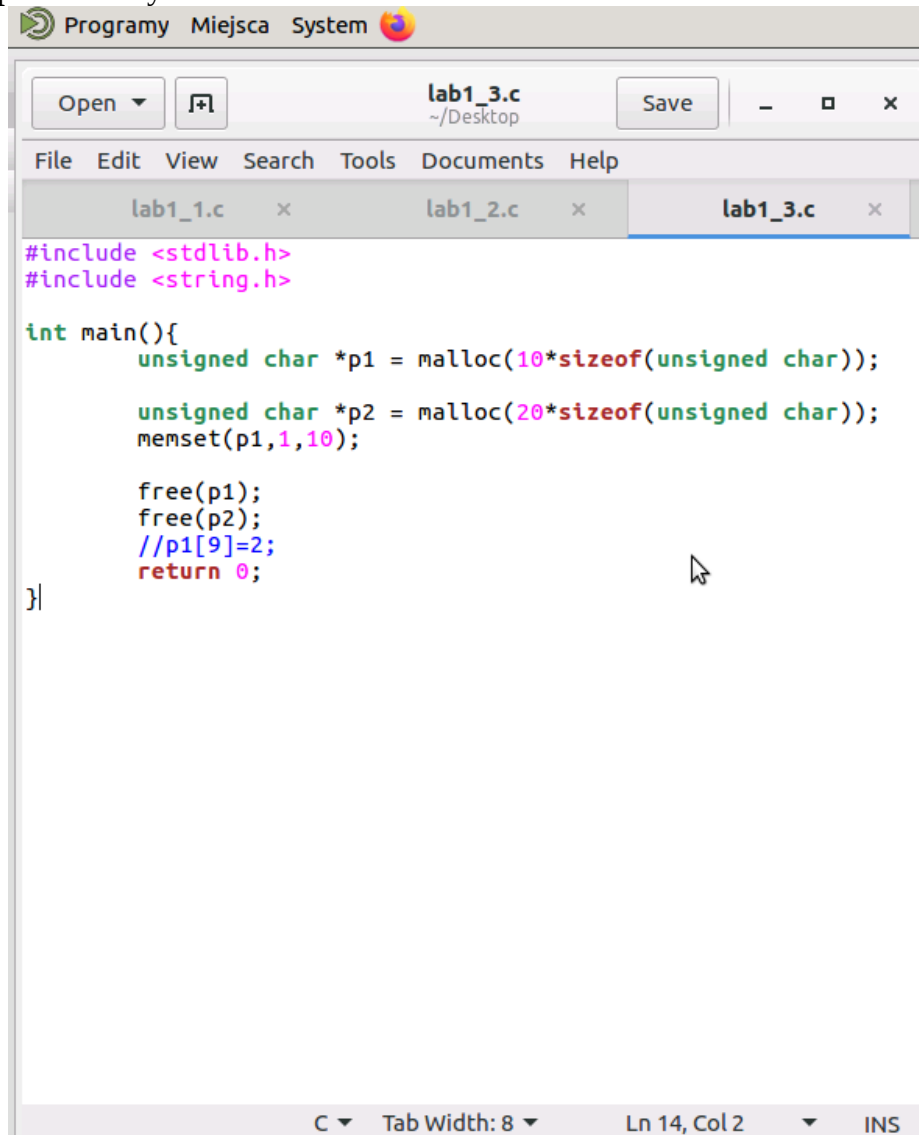
```
Desktop : bash — Konsole
Plik Edycja Widok Bookmarks Ustawienia Pomoc
==9974==    suppressed: 0 bytes in 0 blocks
==9974==
==9974== For counts of detected and suppressed errors, rerun with: -v
==9974== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
bd46477@jota-7267:~/Desktop$ gcc lab1_3.c -g
bd46477@jota-7267:~/Desktop$ valgrind --leak-check=yes ./a.out
==10039== Memcheck, a memory error detector
==10039== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==10039== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==10039== Command: ./a.out
==10039==
==10039== HEAP SUMMARY:
==10039==    in use at exit: 0 bytes in 0 blocks
==10039==    total heap usage: 2 allocs, 2 frees, 30 bytes allocated
==10039==
==10039== All heap blocks were freed -- no leaks are possible
==10039==
==10039== For counts of detected and suppressed errors, rerun with: -v
==10039== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
bd46477@jota-7267:~/Desktop$
```

Desktop : bash

03-04 10:56:14.png Page 3 of 3 64 words, 431 chara

PDF File viewer | Micr... Desktop : bash — Ko... lab1\_3.c (~/Desko

Kod który poprawia błędy:



The screenshot shows a code editor window with a menu bar (File, Edit, View, Search, Tools, Documents, Help) and a toolbar (Open, Save, and window controls). The title bar indicates the file is 'lab1\_3.c' located on the Desktop. The editor contains the following C code:

```
#include <stdlib.h>
#include <string.h>

int main(){
    unsigned char *p1 = malloc(10*sizeof(unsigned char));

    unsigned char *p2 = malloc(20*sizeof(unsigned char));
    memset(p1,1,10);

    free(p1);
    free(p2);
    //p1[9]=2;
    return 0;
}
```

The status bar at the bottom shows 'C', 'Tab Width: 8', 'Ln 14, Col 2', and 'INS'.