Naiwny program

```
#include <stdio.h>

int main(int argc, char **argv)
{
  int i;
  char *buf = NULL;

  if (argc<1)
    {
      fprintf(stderr,"One argument expected...");
      return -1;
    }

  for (i=0; argv[1][i]; i++)
      buf[i]=argv[1][i];   /* Oops.. */

  printf("%s",buf);

  return 0;

}
```

Uruchomienie, kompilacja i debugger

```
$ make test
$ ./test unixic
Segmentation fault (core dumped)

$ gcc -g -c test.c
$ gcc -o test test.o
$ gdb test

GNU gdb 5.0rh-5 Red Hat Linux 7.1
Copyright 2001 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public
License, and you are welcome to change it and/or
distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.
Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux"...

(gdb)
```

Uruchomienie w GDB

```
(gdb) run
Starting program: ./test
One argument expected...
Program exited with code 0377.

(gdb) set args unixic

(gdb) run
Starting program: ./test unixic

Program received signal SIGSEGV, Segmentation fault.
0x08048521 in main (argc=2, argv=0xbffffa7c) at test.c:15
15        buf[i]=argv[1][i];   /* Oops.. */
(gdb)
```

GNU Debugger - pomoc

```
(gdb) help
aliases -- Aliases of other commands
breakpoints -- Making program stop at certain points
data -- Examining data
files -- Specifying and examining files
internals -- Maintenance commands
obscure -- Obscure features
running -- Running the program
stack -- Examining the stack
status -- Status inquiries
tracepoints -- Tracing of program execution without
            stopping the program
user-defined -- User-defined commands
```

GNU Debugger - pomoc

```
(gdb) help breakpoints
List of commands:

awatch -- Set a watchpoint for an expression
break -- Set breakpoint at specified line or function
catch -- Set catchpoints to catch events
clear -- Clear breakpoint at specified line or function
commands -- Set commands to be executed when a breakpoint
         is hit
condition -- Specify breakpoint number N to break only
         if COND is true
delete -- Delete some breakpoints or auto-display
       expressions
disable -- Disable some breakpoints
enable -- Enable some breakpoints
hbreak -- Set a hardware assisted  breakpoint
ignore -- Set ignore-count of breakpoint number N to COUNT
rbreak -- Set a breakpoint for all functions matching
       REGEXP
rwatch -- Set a read watchpoint for an expression
tbreak -- Set a temporary breakpoint
tcatch -- Set temporary catchpoints to catch events
...
(gdb)
```

Sesja debuggera - breakpoints

```
(gdb) list
1 #include <stdio.h>
2
3 int main(int argc, char **argv)
4 {
5   int i;
6   char *buf = NULL;
7
8   if (argc<2)
9     {
10       fprintf(stderr,"One argument expected...");

(gdb) break 6
Breakpoint 1 at 0x80484c6: file test.c, line 6.

(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: ./test unixic

Breakpoint 1, main (argc=2, argv=0xbffffa7c) at test.c:6
6   char *buf = NULL;
```

Sesja debuggera - zmienne, etc...

```
(gdb) display i
1: i = 134518468

(gdb) display argc
2: argc = 2

(gdb) display buf
3: buf = 0x80495e4 ""

(gdb) print argv[1]
$1 = 0xbffffb94 "unixic"

(gdb) display
3: buf = 0x0
2: argc = 2
1: i = 134518468
```

Sesja debuggera - krok po kroku

```
(gdb) next
8   if (argc<2)
3: buf = 0x0
2: argc = 2
1: i = 134518468

(gdb) next
14   for (i=0; argv[1][i]; i++)
3: buf = 0x0
2: argc = 2
1: i = 134518468

(gdb) next
15        buf[i]=argv[1][i];   /* Oops.. */
3: buf = 0x0
2: argc = 2
1: i = 0
```

Sesja debuggera - segmentation fault

```
(gdb) next

Program received signal SIGSEGV, Segmentation fault.
0x08048521 in main (argc=2, argv=0xbffffa7c) at test.c:15
15        buf[i]=argv[1][i];   /* Oops.. */
3: buf = 0x0
2: argc = 2
1: i = 0

(gdb) cont
Continuing.

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.

(gdb) quit
$
```

A co z core?

```
$ ./test unixic
Segmentation fault (core dumped)
$ gdb test core
...
Core was generated by `./test unixic'.
Program terminated with signal 11, Segmentation fault.
Reading symbols from /lib/i686/libc.so.6...done.
Loaded symbols for /lib/i686/libc.so.6
Reading symbols from /lib/ld-linux.so.2...done.
Loaded symbols for /lib/ld-linux.so.2
#0  0x08048521 in main (argc=2, argv=0xbffffa6c) at test.c:15
15        buf[i]=argv[1][i];   /* Oops.. */

(gdb) print i
$1 = 0
(gdb) print argc
$2 = 2
(gdb) print argv[1]
$5 = 0xbffffb94 "unixic"

(gdb)
```