# Программирование в Linux

## Core dumps

```
asdf
x
asdf
x
asdf
x
asdf
x
asdf
x
Segmentation fault (core dumped)
```

# Core file

- Слепок состояния виртуального адресного пространства процесса
- Основное назначение — отладка
- Формат файла не стандартизирован (согласован с версией отладчика)

# Способы генерации

- Автоматически при получении процессом сигнала, требующего дампа
- Вручную, утилитой gcore

| Signal | Standard | Action | Comment |
|---|---|---|---|
| **SIGABRT** | P1990 | Core | Abort signal from abort(3) |
| **SIGALRM** | P1990 | Term | Timer signal from alarm(2) |
| **SIGBUS** | P2001 | Core | Bus error (bad memory access) |
| **SIGCHLD** | P1990 | Ign | Child stopped or terminated |
| **SIGCLD** | - | Ign | A synonym for **SIGCHLD** |
| **SIGCONT** | P1990 | Cont | Continue if stopped |
| **SIGEMT** | - | Term | Emulator trap |
| **SIGFPE** | P1990 | Core | Floating-point exception |
| **SIGHUP** | P1990 | Term | Hangup detected on controlling terminal or death of controlling process |

# Автоматический дамп

- Процессу должно быть разрешено дампить core файлы (RLIMIT_CORE)
- По умолчанию дампится файл с именем core в рабочем каталоге процесса
- В */proc/sys/kernel/core_pattern* можно настроить формат именования и путь сохранения
- Подробнее: `man 5 core`

```c
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int myfunc(int i) {
    *(int*)(NULL) = i; /* line 7 */
    return i - 1;
}

int main(int argc, char **argv) {
    /* Setup some memory. */
    char data_ptr[] = "string in data segment";
    char *mmap_ptr;
    char *text_ptr = "string in text segment";
    (void)argv;
    mmap_ptr = (char *)malloc(sizeof(data_ptr) + 1);
    strcpy(mmap_ptr, data_ptr);
    mmap_ptr[10] = 'm';
    mmap_ptr[11] = 'm';
    mmap_ptr[12] = 'a';
    mmap_ptr[13] = 'p';
    printf("text addr: %p\n", text_ptr);
    printf("data addr: %p\n", data_ptr);
    printf("mmap addr: %p\n", mmap_ptr);

    /* Call a function to prepare a stack trace. */
    return myfunc(argc);
}
```

```
dmis@dmis-MS-7A15:~/LinuxEgs/coredumps$ gcc -g -std=c11 -o fail_test main.c
dmis@dmis-MS-7A15:~/LinuxEgs/coredumps$ ls
fail_test  main.c
dmis@dmis-MS-7A15:~/LinuxEgs/coredumps$ ulimit -c unlimited
dmis@dmis-MS-7A15:~/LinuxEgs/coredumps$ ./fail_test
text addr: 0x56390c94f004
data addr: 0x7ffc783183c0
mmap addr: 0x56390d76a2a0
Segmentation fault (core dumped)
dmis@dmis-MS-7A15:~/LinuxEgs/coredumps$ ls
core  fail_test  main.c
```

```
dmis@dmis-MS-7A15:~/LinuxEgs/coredumps$ gdb -q -nh fail_test core
Reading symbols from fail_test...
[New LWP 492417]
Core was generated by `./fail_test'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x000056390c94e1bc in myfunc (i=1) at main.c:7
7           *(int*)(NULL) = i; /* line 7 */
```

# Дамп вручную

- gcore pid
- сдампит core указанного запущенного процесса в текущий каталог
- процесс останется жить и работать