Exercice 1

strace echo "Hello World"

```
execve("/usr/bin/echo", ["echo", "Hello World"], 0x7fff1b40be68 /* 48 vars */) = 0
```

## used for executing a program; using as arguments the path ("/usr/bin/echo"), the command that we want (["echo", "Hello World"]) and the environment of the program (0x7fff1b40be68). Returning 0 means it was a success.

```
brk(NULL)                        = 0x559c5ff2d000
```

## used to make a break in the process; the value 0x559c5ff2d000 means that the allocated memory for the next address is going to be resume

```
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffea6ae5e80) = -1 EINVAL (Argument invalide)
```

## set an architecture process or a thread state. 0x3001 is the selected subfunction and 0x7ffea6ae5e80 its adress. Returning -1 EINVAL means that the selected subfunction is not valid.

```
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## check if the calling process has access to this filename path "/etc/ld.so.preload", R_OK means that it is trying to read. Returning -1 ENOENT means an error and that such a file doesn’t exist.

```
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/ld.so.cache" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close on exec. Returning 3 is the file descriptor (and it was a success).

```
fstat(3, {st_mode=S_IFREG|0644, st_size=80743, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=80743, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 80743, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f98ba37a000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 80743 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE mens that the mapping is going to private (prevents other for using it), 3 is still the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libc.so.6" to the flag O_RDONLY|O_CLOEXEC only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

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
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

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
## attempts to read up to 784 bytes from file descriptor 3 at offset 64 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

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
## attempts to read up to 32 bytes from file descriptor 3 at offset 848 into the buffer starting at "\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”. Number of bytes is returned, so it was a success.

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
## attempts to read up to 68 bytes from file descriptor 3 at offset 880 into the buffer starting at "\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”. Number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=2029224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f98ba378000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE|MAP_ANONYMOUS means that the mapping is going to private (prevents others from using it) and all the value

is resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

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
## attempts to read up to 784 bytes from file descriptor 3 at offset 64 into the buffer starting at "\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”. Number of bytes is returned, so it was a success.

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
## attempts to read up to 784 bytes from file descriptor 3 at offset 64 into the buffer starting at "\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”. Number of bytes is returned, so it was a success.

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
## attempts to read up to 68 bytes from file descriptor 3 at offset 880 into the buffer starting at "\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”. Number of bytes is returned, so it was a success.

mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f98ba186000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 2036952 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mprotect(0x7f98ba1ab000, 1847296, PROT_NONE) = 0
## changes the access protections for the calling process's memory. 0x7f98ba1ab000 is the address of a region in the memory, 1847296 is the size of the protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

mmap(0x7f98ba1ab000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f98ba1ab000
## creates a new map file to memory. 0x7f98ba1ab000 represents the address was inputted for the map location, 1540096 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place

exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x25000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7f98ba323000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f98ba323000
## creates a new map file to memory. 0x7f98ba323000 represents the address was inputted for the map location, 303104 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x19d000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7f98ba36e000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f98ba36e000
## creates a new map file to memory. 0x7f98ba36e000 represents the address was inputted for the map location, 24576 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x1e7000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7f98ba374000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f98ba374000
## creates a new map file to memory. 0x7f98ba374000 represents the address was inputted for the map location, 13528 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

arch_prctl(ARCH_SET_FS, 0x7f98ba379580) = 0
## set an architecture process or a thread state. ARCH_SET_FS set the 64bit base for the FS register to 0x7f98ba379580 address. Returning 0 means it was a success.

mprotect(0x7f98ba36e000, 12288, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7f98ba36e000 is the address of a region in the memory, 12288 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x559c5fdc4000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x559c5fdc4000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7f98ba3bb000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7f98ba3bb000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

munmap(0x7f98ba37a000, 80743)   = 0
## deletes the address for the specified address region, and unmapps it. 0x7f98ba37a000 is the specified address region and 80743 its length (not necessarily). Return 0 meaning it was a success.

brk(NULL)   = 0x559c5ff2d000
## used to make a break in the process; the value 0x559c5ff2d000 means that the allocated memory for the next address is going to be resumed.

brk(0x559c5ff4e000)   = 0x559c5ff4e000
## used to make a break in the process; the value 0x559c5ff4e000 means that the allocated memory for the next address is going to be resumed.

openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path ""/usr/lib/locale/locale-archive" to the flag O_RDONLY|O_CLOEXEC only for reading and enabling the close on exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=14537584, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=14537584, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 14537584, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f98b93a8000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 14537584 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE mens that the mapping is going to private (prevents other for using it), 3 is still the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                        = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
```

## obtain information about the file descriptor 1, {st_mode=S_IFREG|0620, st_rdev=makedev(0x88, 0), ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
write(1, "Hello World\n", 12Hello World)          = 12
```

## writes up to 12 bytes "Hello World\n" to the file descriptor 1. The return value is the number of bytes.

```
close(1)                        = 0
```

## close the file descriptor 1. Return 0 means it was a success.

```
close(2)                        = 0
```

## close the file descriptor 2. Return 0 means it was a success.

```
exit_group(0)                   = ?
```

## exit all threads in the calling process. End of the strace.

```
        +++ exited with 0 +++
```

---

strace ls -la

```
execve("/usr/bin/ls", ["ls", "-la"], 0x7ffe95b27d08 /* 48 vars */) = 0
```

## used for executing a program; using as arguments the path ("/usr/bin/ls"), the command that we want (["ls", "-la"]) and the environment of the program (0x7ffe95b27d08). Returning 0 means it was a success.

```
brk(NULL)                       = 0x55a0aef49000
```

## used to make a break in the process; the value 0x55a0aef49000 means that the allocated memory for the next address is going to be resume

```
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe19824480) = -1 EINVAL (Argument invalide)
```

## set an architecture process or a thread state. 0x3001 is the selected subfunction and 0x7ffe19824480 its address. Returning -1 EINVAL means that the selected subfunction is not valid.

```
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## check if the calling process has access to this filename path "/etc/ld.so.preload", R_OK means that it is trying to read. Returning -1 ENOENT means an error and that such a file doesn’t exist.

```
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/ld.so.cache" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

```
fstat(3, {st_mode=S_IFREG|0644, st_size=80743, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=80743, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 80743, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb74bcdb000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 80743 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE mens that the mapping is going to private (prevents other for using it), 3 is still the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                          = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libselinux.so.1" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close on exec. Returning 3 is the file descriptor (and it was a success).

```
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@p\0\0\0\0\0\0"..., 832) = 832
```

## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@p\0\0\0\0\0\0". Number of bytes is returned, so it was a success.

```
fstat(3, {st_mode=S_IFREG|0644, st_size=163200, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=2029224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fb74bcd9000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE|MAP_ANONYMOUS means that the mapping is going to private (prevents others from using it) and all the value is resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mmap(NULL, 174600, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb74bcae000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 174600 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mprotect(0x7fb74bcb4000, 135168, PROT_NONE) = 0
## changes the access protections for the calling process's memory. 0x7fb74bcb4000 is the address of a region in the memory, 135168 is the size of the protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

mmap(0x7fb74bcb4000, 102400, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7fb74bcb4000
## creates a new map file to memory. 0x7fb74bcb4000 represents the address was inputted for the map location, 102400 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x6000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74bccd000, 28672, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f000) = 0x7fb74bccd000
## creates a new map file to memory. 0x7fb74bccd000 represents the address was inputted for the map location, 28672 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we

can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x1f000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74bcd5000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7fb74bcd5000
## creates a new map file to memory. 0x7fb74bcd5000 represents the address was inputted for the map location, 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x26000means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74bcd7000, 6664, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb74bcd7000
## creates a new map file to memory. 0x7fb74bcd7000 represents the address was inputted for the map location, 6664 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libc.so.6" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

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
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

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
## attempts to read up to 784 bytes from file descriptor 3 at offset 64 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

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
## attempts to read up to 32 bytes from file descriptor 3 at offset 848 into the buffer starting at "\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”. Number of bytes is returned, so it was a success.

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
## attempts to read up to 68 bytes from file descriptor 3 at offset 880 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=2029224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

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
## attempts to read up to 784 bytes from file descriptor 3 at offset 64 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

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
## attempts to read up to 32 bytes from file descriptor 3 at offset 848 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

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
## attempts to read up to 68 bytes from file descriptor 3 at offset 880 into the buffer starting at "\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". Number of bytes is returned, so it was a success.

mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb74babc000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 2036952 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mprotect(0x7fb74bae1000, 1847296, PROT_NONE) = 0
## changes the access protections for the calling process's memory. 0x7fb74bae1000 is the address of a region in the memory, 1847296 is the size of the protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

mmap(0x7fb74bae1000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7fb74bae1000
## creates a new map file to memory. 0x7fb74bae1000 represents that an address was inputted for the map location, 1540096 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x25000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74bc59000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7fb74bc59000
## creates a new map file to memory. 0x7fb74bc59000 represents that an address was inputted for the map location, 303104 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x19d000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74bca4000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fb74bca4000
## creates a new map file to memory. 0x7fb74bca4000 represents that an address was inputted for the map location, 24576 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x1e7000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74bcaa000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb74bcaa000
## creates a new map file to memory. 0x7fb74bcaa000 represents that an address was inputted for the map location, 13528 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we

can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3) = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcre2-8.so.0", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libpcre2-8.so.0" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\340\"\0\0\0\0\0\0"..., 832) = 832
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\340\"\0\0\0\0\0\0". The number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0644, st_size=584392, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=584392, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 586536, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb74ba2c000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 586536 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mmap(0x7fb74ba2e000, 409600, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fb74ba2e000
## creates a new map file to memory. 0x7fb74ba2e000 represents that an address was inputted for the map location, 409600 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file

descriptor and 0x2000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ba92000, 163840, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x66000) = 0x7fb74ba92000
## creates a new map file to memory. 0x7fb74ba92000 represents that an address was inputted for the map location, 163840 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x66000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74baba000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8d000) = 0x7fb74baba000
## creates a new map file to memory. 0x7fb74baba000 represents that an address was inputted for the map location, 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x8d000 means that there is an offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libdl.so.2" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 \22\0\0\0\0\0\0"..., 832) = 832
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 \22\0\0\0\0\0\0". Number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0644, st_size=18816, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=18816, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb74ba26000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 20752 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mmap(0x7fb74ba27000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fb74ba27000
## creates a new map file to memory. 0x7fb74ba27000 represents that an address was inputted for the map location, 8192 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x1000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ba29000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fb74ba29000
## creates a new map file to memory. 0x7fb74ba29000, represents that an address was inputted for the map location, 4096 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x3000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ba2a000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fb74ba2a000
## creates a new map file to memory. 0x7fb74ba2a000, represents that an address was inputted for the map location, 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x3000 means that there is an offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libpthread.so.0" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0\0"..., 832) = 832
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0\0". The number of bytes is returned, so it was a success.

pread64(3,"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68
## attempts to read up to 68 bytes from file descriptor 3 at offset 824 into the buffer starting at "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337". Number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=157224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

pread64(3,"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68
## attempts to read up to 68 bytes from file descriptor 3 at offset 824 into the buffer starting at "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337". Number of bytes is returned, so it was a success.

mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb74ba03000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 140408 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mmap(0x7fb74ba0a000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7fb74ba0a000
## creates a new map file to memory. 0x7fb74ba0a000, represents that an address was inputted for the map location, 69632 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x7000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ba1b000, 20480, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7fb74ba1b000
## creates a new map file to memory. 0x7fb74ba1b000, represents that an address was inputted for the map location, 20480 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x18000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ba20000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7fb74ba20000
## creates a new map file to memory. 0x7fb74ba20000, represents that an address was inputted for the map location, 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0x1c000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ba22000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb74ba22000
## creates a new map file to memory. 0x7fb74ba22000, represents that an address was inputted for the map location, 13432 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fb74ba01000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

arch_prctl(ARCH_SET_FS, 0x7fb74ba02400) = 0
## set an architecture process or a thread state. ARCH_SET_FS set the 64bit base for the FS register to 0x7fb74ba02400 address. Returning 0 means it was a success.

mprotect(0x7fb74bca4000, 12288, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb74bca4000 is the address of a region in the memory, 12288 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb74ba20000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb74ba20000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb74ba2a000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb74ba2a000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb74baba000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb74baba000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb74bcd5000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb74bcd5000 is the address of a region in the memory, 4096 is the size of the

protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x55a0aea28000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x55a0aea28000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb74bd1c000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb74bd1c000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

munmap(0x7fb74bcdb000, 80743)   = 0
## deletes the address for the specified address region, and unmapps it. 0x7fb74bcdb000 is the specified address region and 80743 its length (not necessarily). Return 0 meaning it was a success.

set_tid_address(0x7fb74ba026d0)   = 5023
## sets the clear_child_tid value for the calling thread to 0x7fb74ba026d0. Return the caller's thread ID.

set_robust_list(0x7fb74ba026e0, 24)   = 0
## informs the kernel of the location of its robust futex list, it is used in case a thread fails to unlock a futex before terminating. 0x7fb74ba026e0 means the kernel to record the head of the list of robust futexes owned by the calling thread in a size of 24. Return 0 means it was a success.

rt_sigaction(SIGRTMIN, {sa_handler=0x7fb74ba0abf0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7fb74ba183c0}, NULL, 8) = 0
## used to change the action taken by a process on receipt of a specific signal. This new action {sa_handler=0x7fb74ba0abf0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7fb74ba183c0} for signal SIGRTMIN is saved. 8 is the size in bytes for sa_mask. Return 0 on success.

rt_sigaction(SIGRT_1, {sa_handler=0x7fb74ba0ac90, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7fb74ba183c0}, NULL, 8) = 0
## used to change the action taken by a process on receipt of a specific signal. This new action {sa_handler=0x7fb74ba0ac90, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO,

sa_restorer=0x7fb74ba183c0} for signal SIGRT_1 is saved. 8 is the size in bytes for sa_mask. Return 0 on success.

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
## used to fetch and/or change the signal mask of the calling thread. Since SIG_UNBLOCK then the signals in [RTMIN RT_1] set are removed from the current set of blocked signals. 8 is the size in bytes for [RTMIN RT_1]. Return 0 on success.

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
## get and set resource limits. 0 is the process ID(here it is the calling proccess), RLIMIT_STACK is the maximum size of the process stack in bytes, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY} is the range of the ressources. Return 0 on success.

statfs("/sys/fs/selinux", 0x7ffe198243d0) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## returns information about a mounted filesystem. "/sys/fs/selinux" is the pathname of any file within the mounted filesystem. 0x7ffe198243d0 is a pointer to a statfs structure defined. Return -1 is an error where the path doesn’t exist.

statfs("/selinux", 0x7ffe198243d0)   = -1 ENOENT (Aucun fichier ou dossier de ce type)
## returns information about a mounted filesystem. "/selinux"" is the pathname of any file within the mounted filesystem. 0x7ffe198243d0 is a pointer to a statfs structure defined. Return -1 is an error where the path doesn’t exist.

brk(NULL)   = 0x55a0aef49000
## used to make a break in the process; the value 0x55a0aef49000 means that the allocated memory for the next address is going to be resumed.

brk(0x55a0aef6a000)   = 0x55a0aef6a000
## used to make a break in the process; the value 0x55a0aef6a000 means that the allocated memory for the next address is going to be resumed.

openat(AT_FDCWD, "/proc/filesystems", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/proc/filesystems" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0444, st_size=0, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
read(3, "nodev\tsysfs\nnodev\ttmpfs\nnodev\tbd"..., 1024) = 440
```

## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "nodev\tsysfs\nnodev\ttmpfs\nnodev\tbd". A number smaller than the number of bytes is returned, so it is still a success.

```
read(3, "", 1024)                 = 0
```

## attempts to read up to 1024 bytes from file descriptor 3 into the buffer starting at "". Return 0 meaning it is the end of the file.

```
close(3)                          = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
access("/etc/selinux/config", F_OK)     = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## check if the calling process has access to this filename path "/etc/selinux/config", F_OK means that it is trying to find if such a file exists. Returning -1 ENOENT means an error and that such a file doesn't exist.

```
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/lib/locale/locale-archive" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

```
fstat(3, {st_mode=S_IFREG|0644, st_size=14537584, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=14537584, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 14537584, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb74ac23000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 14537584 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                          = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
```

## manipulates the underlying device parameters of special files. 1 is the file descriptor, TCGETS gets the current serial port settings. Return 0 on success.

ioctl(1, TIOCGWINSZ, {ws_row=24, ws_col=80, ws_xpixel=0, ws_ypixel=0}) = 0
## manipulates the underlying device parameters of special files. 1 is the file descriptor, TIOCGWINSZ gets window size. Return 0 on success.

openat(AT_FDCWD, "/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/locale.alias" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=2996, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=2996, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

read(3, "# Locale name alias data base.\n#"..., 4096) = 2996
## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "# Locale name alias data base.\n#". A number smaller than the number of bytes is returned, so it is still a success.

read(3, "", 4096)   = 0
## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "". Return 0 meaning it is the end of the file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/usr/share/locale/fr_FR.UTF-8/LC_TIME/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr_FR.UTF-8/LC_TIME/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale/fr_FR.utf8/LC_TIME/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr_FR.utf8/LC_TIME/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale/fr_FR/LC_TIME/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr_FR/LC_TIME/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn't exist.

openat(AT_FDCWD, "/usr/share/locale/fr.UTF-8/LC_TIME/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr.UTF-8/LC_TIME/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn't exist.

openat(AT_FDCWD, "/usr/share/locale/fr.utf8/LC_TIME/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr.utf8/LC_TIME/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn't exist.

openat(AT_FDCWD, "/usr/share/locale/fr/LC_TIME/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr/LC_TIME/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn't exist.

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache", O_RDONLY) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache" to the flag O_RDONLY means only for reading. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=27002, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=27002, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 27002, PROT_READ, MAP_SHARED, 3, 0) = 0x7fb74bce8000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 27002 is the length of the mapping, PROT_READ is the protection of the memory and it means that we

can read it, MAP_SHARED shares this mapping, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                    = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
futex(0x7fb74bca9634, FUTEX_WAKE_PRIVATE, 2147483647) = 0
```

## provides a method for waiting until a certain condition becomes true. 0x7fb74bca9634 is the futex word, FUTEX_WAKE_PRIVATE means it wakes at 2147483647. Return 0 on success.

```
openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "." to the flag O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY means only for reading, opening the file in nonblocking mode, enabling the close-on-exec and the path "." is not a directory. Returning 3 is the file descriptor (and it was a success).

```
fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
getdents64(3, /* 28 entries */, 32768)  = 896
```

## gets directory entries on file descriptor 3 into buffer pointed to by /* 28 entries */. 32768 is the size of this buffer. Return the number of bytes read.

```
lstat("Public", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
```

## obtain information about the file pointed to by path "Public", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
lgetxattr("Public", "security.selinux", 0x55a0aef4b2f0, 255) = -1 ENODATA (Aucune
donnée disponible)
```

## retrieves the value 0x55a0aef4b2f0 of the extended attribute identified by "security.selinux" and associated with "Public" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

```
getxattr("Public", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune
donnée disponible)
```

## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Public" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("Public", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Public" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 4
## creates an endpoint for communication. The communication domain is AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK provides sequenced connection-based byte streams, sets the close-on-exec flag on the new fd and sets the O_NONBLOCK file status flag on the open fd. Return the new fd.

connect(4, {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"}, 110) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## connects the socket referred to by the file descriptor 4 to the address {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"} with a size of 110. Return -1 error, such a file doesn't exist.

close(4)   = 0
## close the file descriptor 4. Return 0 means it was a success.

socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 4
## creates an endpoint for communication. The communication domain is AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK provides sequenced connection-based byte streams, sets the close-on-exec flag on the new fd and sets the O_NONBLOCK file status flag on the open fd. Return -1 an error, such a file doesn't exist.

connect(4, {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"}, 110) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## connects the socket referred to by the file descriptor 4 to the address {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"} with a size of 110. Return -1 error, such a file doesn't exist.

close(4)   = 0
## close the file descriptor 4. Return 0 means it was a success.

openat(AT_FDCWD, "/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 4
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/nsswitch.conf" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

fstat(4, {st_mode=S_IFREG|0644, st_size=542, ...}) = 0
## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=542, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

read(4, "# /etc/nsswitch.conf\n#\n# Example"..., 4096) = 542
## attempts to read up to 4096 bytes from file descriptor 4 into the buffer starting at "# /etc/nsswitch.conf\n#\n# Example". A number smaller than the number of bytes is returned, so it is still a success.

read(4, "", 4096)   = 0
## attempts to read up to 4096 bytes from file descriptor 4 into the buffer starting at "". Return 0 meaning it is the end of the file.

close(4)   = 0
## close the file descriptor 4. Return 0 means it was a success.

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 4
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/ld.so.cache" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

fstat(4, {st_mode=S_IFREG|0644, st_size=80743, ...}) = 0
## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=80743, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 80743, PROT_READ, MAP_PRIVATE, 4, 0) = 0x7fb74ac0f000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 80743 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), 4 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(4)   = 0
## close the file descriptor 4. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libnss_files.so.2", O_RDONLY|O_CLOEXEC) = 4
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libnss_files.so.2" to the flag O_RDONLY|O_CLOEXEC means only for reading

and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

read(4, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3005\0\0\0\0\0\0"..., 832) = 832
## attempts to read up to 832 bytes from file descriptor 4 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3005\0\0\0\0\0\0". The number of bytes is returned, so it is a success.

fstat(4, {st_mode=S_IFREG|0644, st_size=51832, ...}) = 0
## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=51832, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 79672, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 4, 0) = 0x7fb74abfb000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 79672 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 4 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mmap(0x7fb74abfe000, 28672, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x3000) = 0x7fb74abfe000
## creates a new map file to memory. 0x7fb74abfe000 represents that an address was inputted for the map location, 28672 is the length of the mapping, PROT_READ|PROT_EXEC is the protection of the memory and it means that we can read and execute it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 4 is the file descriptor and 0x3000 means that there is an offset. The returned value is the address of the map file.

mmap(0x7fb74ac05000, 8192, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0xa000) = 0x7fb74ac05000
## creates a new map file to memory. 0x7fb74ac05000 represents that an address was inputted for the map location, 8192 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 4 is the file descriptor and 0xa000 means that there is an offset. The returned value is the address of the map file.

```
mmap(0x7fb74ac07000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0xb000) = 0x7fb74ac07000
```

## creates a new map file to memory. 0x7fb74ac07000 represents that an address was inputted for the map location, 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and we can ignore the MAP_DENYWRITE flag, 4 is the file descriptor and 0xb000 means that there is an offset. The returned value is the address of the map file.

```
mmap(0x7fb74ac09000, 22328, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb74ac09000
```

## creates a new map file to memory. 0x7fb74ac09000 represents that an address was inputted for the map location, 22328 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), MAP_FIXED means that the map is going to be place exactly at this address and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(4)                    = 0
```

## close the file descriptor 4. Return 0 means it was a success.

```
mprotect(0x7fb74ac07000, 4096, PROT_READ) = 0
```

## changes the access protections for the calling process's memory. 0x7fb74ac07000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

```
munmap(0x7fb74ac0f000, 80743)       = 0
```

## deletes the address for the specified address region, and unmapps it. 0x7fb74ac0f000 is the specified address region and 80743 its length (not necessarily). Return 0 meaning it was a success.

```
openat(AT_FDCWD, "/etc/passwd", O_RDONLY|O_CLOEXEC) = 4
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/passwd" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

lseek(4, 0, SEEK_CUR) = 0
## used to change the location of the read/write pointer of a fd. 4 is the current fd, 0 is the offset and SEEK_CUR means the offset is set to 0 bytes. Return the offset 0.

fstat(4, {st_mode=S_IFREG|0644, st_size=2746, ...}) = 0
## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=2746, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

read(4, "root:x:0:0:root:/root:/bin/bash\n"..., 4096) = 2746
## attempts to read up to 832 bytes from file descriptor 4 into the buffer starting at "root:x:0:0:root:/root:/bin/bash\n". A smaller number of bytes is returned, so it is still a success.

close(4) = 0
## close the file descriptor 4. Return 0 means it was a success.

socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 4
## creates an endpoint for communication. The communication domain is AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK provides sequenced connection-based byte streams, sets the close-on-exec flag on the new fd and sets the O_NONBLOCK file status flag on the open fd. Return the new fd.

connect(4, {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"}, 110) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## connects the socket referred to by the file descriptor 4 to the address {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"} with a size of 110. Return -1 error, such a file doesn't exist.

close(4) = 0
## close the file descriptor 4. Return 0 means it was a success.

socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 4
## creates an endpoint for communication. The communication domain is AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK provides sequenced connection-based byte streams, sets the close-on-exec flag on the new fd and sets the O_NONBLOCK file status flag on the open fd. Return the new fd.

connect(4, {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"}, 110) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## connects the socket referred to by the file descriptor 4 to the address {sa_family=AF_UNIX, sun_path="/var/run/nscd/socket"} with a size of 110. Return -1 error, such a file doesn't exist.

```
close(4)                        = 0
```

## close the file descriptor 4. Return 0 means it was a success.

```
openat(AT_FDCWD, "/etc/group", O_RDONLY|O_CLOEXEC) = 4
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/group" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

```
lseek(4, 0, SEEK_CUR)           = 0
```

## used to change the location of the read/write pointer of a fd. 4 is the current fd, 0 is the offset and SEEK_CUR means the offset is set to 0 bytes. Return the offset 0.

```
fstat(4, {st_mode=S_IFREG|0644, st_size=1048, ...}) = 0
```

## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=1048, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
read(4, "root:x:0:\ndaemon:x:1:\nbin:x:2:\ns"..., 4096) = 1048
```

## attempts to read up to 832 bytes from file descriptor 4 into the buffer starting at "root:x:0:\ndaemon:x:1:\nbin:x:2:\ns". A smaller number of bytes is returned, so it is still a success.

```
close(4)                        = 0
```

## close the file descriptor 4. Return 0 means it was a success.

```
lstat(".gnupg", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
```

## obtain information about the file pointed to by path ".gnupg", {st_mode=S_IFDIR|0700, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
lgetxattr(".gnupg", "security.selinux", 0x55a0aef5a2f0, 255) = -1 ENODATA (Aucune donnée disponible)
```

## retrieves the value 0x55a0aef5a2f0 of the extended attribute identified by "security.selinux" and associated with ".gnupg" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

```
getxattr(".gnupg", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
```

## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".gnupg" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".gnupg", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".gnupg" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".steampath", {st_mode=S_IFLNK|0777, st_size=30, ...}) = 0
## obtain information about the file pointed to by path ".steampath", {st_mode=S_IFLNK|0777, st_size=30, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".steampath", "security.selinux", 0x55a0aef5a420, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5a420 of the extended attribute identified by "security.selinux" and associated with ".steampath" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

readlink(".steampath", "/home/zheng/.steam/sdk32/steam", 31) = 30
## places the contents of the symbolic link ".steampath" in the buffer "/home/zheng/.steam/sdk32/steam", which has size 31. Return the size of the buffer.

lstat("snap", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "snap", {st_mode=S_IFDIR|0700, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("snap", "security.selinux", 0x55a0aef5a5b0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5a5b0 of the extended attribute identified by "security.selinux" and associated with "snap" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr("snap", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "snap" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("snap", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "snap" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".local", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".local", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".local", "security.selinux", 0x55a0aef5a6e0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5a6e0 of the extended attribute identified by "system.selinux" and associated with ".local" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr(".local", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".local" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

getxattr(".local", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".local" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat(".bashrc", {st_mode=S_IFREG|0644, st_size=3771, ...}) = 0
## obtain information about the file pointed to by path ".bashrc", {st_mode=S_IFREG|0644, st_size=3771, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".bashrc", "security.selinux", 0x55a0aef5a810, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5a810 of the extended attribute identified by "security.selinux" and associated with ".bashrc" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr(".bashrc", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".bashrc" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat("Vid\303\251os", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Vid\303\251os", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Vid\303\251os", "security.selinux", 0x55a0aef5a940, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5a940 of the extended attribute identified by "security.selinux" and associated with "Vid\303\251os" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr("Vid\303\251os", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Vid\303\251os" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("Vid\303\251os", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## obtain information about the file pointed to by path ".", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lstat(".", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".", "security.selinux", 0x55a0aef5aa70, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5aa70 of the extended attribute identified by "security.selinux" and associated with "." in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "." in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "." in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat("Téléchargements", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Téléchargements", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Téléchargements", "security.selinux", 0x55a0aef5aa90, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5aa90 of the extended attribute identified by "security.selinux" and associated with "Téléchargements" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr("Téléchargements", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Téléchargements" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("Téléchargements", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Téléchargements" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".config", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".config", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".config", "security.selinux", 0x55a0aef5aab0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5aab0 of the extended attribute identified by "security.selinux" and associated with ".config" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".config", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".config" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".config", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".config" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".mozilla", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".mozilla", {st_mode=S_IFDIR|0700, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".mozilla", "security.selinux", 0x55a0aef5aad0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5aad0 of the extended attribute identified by "security.selinux" and associated with ".mozilla" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr(".mozilla", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".mozilla" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

getxattr(".mozilla", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5aad0 of the extended attribute identified by "system.posix_acl_default" and associated with ".mozilla" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat(".bash_logout", {st_mode=S_IFREG|0644, st_size=220, ...}) = 0
## obtain information about the file pointed to by path ".bash_logout", {st_mode=S_IFREG|0644, st_size=220, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".bash_logout", "security.selinux", 0x55a0aef5aaf0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5aaf0 of the extended attribute identified by "security.selinux" and associated with ".bash_logout" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr(".bash_logout", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".bash_logout" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat("Documents", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Documents", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Documents", "security.selinux", 0x55a0aef5ab10, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ab10 of the extended attribute identified by "security.selinux" and associated with "Documents" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr("Documents", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Documents" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("Documents", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Documents" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".sudo_as_admin_successful", {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
## obtain information about the file pointed to by path ".sudo_as_admin_successful", {st_mode=S_IFREG|0644, st_size=0, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".sudo_as_admin_successful", "security.selinux", 0x55a0aef5ab30, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ab30 of the extended attribute identified by "security.selinux" and associated with ".sudo_as_admin_successful" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".sudo_as_admin_successful", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".sudo_as_admin_successful" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

lstat(".pki", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".pki", {st_mode=S_IFDIR|0700, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".pki", "security.selinux", 0x55a0aef5ab30, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ab30 of the extended attribute identified by "security.selinux" and associated with ".pki" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".pki", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".pki" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".pki", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".pki" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".steam", {st_mode=S_IFDIR|0775, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".steam", {st_mode=S_IFDIR|0775, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".steam", "security.selinux", 0x55a0aef5ab50, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ab50 of the extended attribute identified by "security.selinux" and associated with ".steam" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".steam", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".steam" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".steam", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".steam" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".cache", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".cache", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".cache", "security.selinux", 0x55a0aef5ab70, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ab70 of the extended attribute identified by "security.selinux" and associated with ".cache" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".cache", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".cache" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".cache", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".cache" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".ssh", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path ".ssh", {st_mode=S_IFDIR|0700, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".ssh", "security.selinux", 0x55a0aef5ab90, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ab90 of the extended attribute identified by "security.selinux" and associated with ".ssh" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".ssh", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".ssh" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr(".ssh", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".ssh" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat("Musique", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Musique", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Musique", "security.selinux", 0x55a0aef5abb0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5abb0 of the extended attribute identified by "security.selinux" and associated with "Musique" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr("Musique", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Musique" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("Musique", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Musique" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".pulse-cookie", {st_mode=S_IFREG|0600, st_size=256, ...}) = 0
## obtain information about the file pointed to by path ".pulse-cookie", {st_mode=S_IFREG|0600, st_size=256, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".pulse-cookie", "security.selinux", 0x55a0aef5abd0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5abd0 of the extended attribute identified by "security.selinux" and associated with ".pulse-cookie" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".pulse-cookie", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".pulse-cookie" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat(".profile", {st_mode=S_IFREG|0644, st_size=807, ...}) = 0
## obtain information about the file pointed to by path ".profile", {st_mode=S_IFREG|0644, st_size=807, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".profile", "security.selinux", 0x55a0aef5abf0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5abf0 of the extended attribute identified by "security.selinux" and associated with ".profile" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr(".profile", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".profile" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

lstat("..", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "..", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("..", "security.selinux", 0x55a0aef5ac10, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ac10 of the extended attribute identified by "security.selinux" and associated with ".." in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

getxattr("..", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".." in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

getxattr("..", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with ".." in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

openat(AT_FDCWD, "/etc/passwd", O_RDONLY|O_CLOEXEC) = 4
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/passwd" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

lseek(4, 0, SEEK_CUR)   = 0
## used to change the location of the read/write pointer of a fd. 4 is the current fd, 0 is the offset and SEEK_CUR means the offset is set to 0 bytes. Return the offset 0.

fstat(4, {st_mode=S_IFREG|0644, st_size=2746, ...}) = 0
## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=2746, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

read(4, "root:x:0:0:root:/root:/bin/bash\n"..., 4096) = 2746
## attempts to read up to 4096 bytes from file descriptor 4 into the buffer starting at "root:x:0:0:root:/root:/bin/bash\n". A smaller number of bytes is returned, so it is still a success.

close(4)   = 0
## close the file descriptor 4. Return 0 means it was a success.

openat(AT_FDCWD, "/etc/group", O_RDONLY|O_CLOEXEC) = 4
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/group" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 4 is the file descriptor (and it was a success).

lseek(4, 0, SEEK_CUR)   = 0
## used to change the location of the read/write pointer of a fd. 4 is the current fd, 0 is the offset and SEEK_CUR means the offset is set to 0 bytes. Return the offset 0.

fstat(4, {st_mode=S_IFREG|0644, st_size=1048, ...}) = 0
## obtain information about the file descriptor 4, {st_mode=S_IFREG|0644, st_size=1048, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

read(4, "root:x:0:\ndaemon:x:1:\nbin:x:2:\ns"..., 4096) = 1048
## attempts to read up to 4096 bytes from file descriptor 4 into the buffer starting at "root:x:0:\ndaemon:x:1:\nbin:x:2:\ns". A smaller number of bytes is returned, so it is still a success.

close(4)   = 0
## close the file descriptor 4. Return 0 means it was a success.

lstat(".steampid", {st_mode=S_IFLNK|0777, st_size=28, ...}) = 0
## obtain information about the file pointed to by path ".steampid", {st_mode=S_IFLNK|0777, st_size=28, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".steampid", "security.selinux", 0x55a0aef5ac70, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ac70 of the extended attribute identified by "security.selinux" and associated with ".steampid" in the file system. The length of the attribute value is 255. Return -1, error data couldn’t be found.

readlink(".steampid", "/home/zheng/.steam/steam.pid", 29) = 28
## places the contents of the symbolic link ".steampath" in the buffer "/home/zheng/.steam/steam.pid", which has size 29. Return the size of the buffer.

lstat("Bureau", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Bureau", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Bureau", "security.selinux", 0x55a0aef5acf0, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5acf0 of the extended attribute identified by "security.selinux" and associated with "Bureau" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr("Bureau", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Bureau" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

getxattr("Bureau", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Bureau" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat("Mod\303\250les", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Mod\303\250les", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Mod\303\250les", "security.selinux", 0x55a0aef5ad10, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ad10 of the extended attribute identified by "security.selinux" and associated with "Mod\303\250les" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr("Mod\303\250les", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Mod\303\250les" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

getxattr("Mod\303\250les", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Mod\303\250les" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat(".wget-hsts", {st_mode=S_IFREG|0664, st_size=177, ...}) = 0
## obtain information about the file pointed to by path ".wget-hsts", {st_mode=S_IFREG|0664, st_size=177, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".wget-hsts", "security.selinux", 0x55a0aef5ad30, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ad30 of the extended attribute identified by "security.selinux" and associated with ".wget-hsts" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr(".wget-hsts", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".wget-hsts" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat(".bash_history", {st_mode=S_IFREG|0600, st_size=2295, ...}) = 0
## obtain information about the file pointed to by path ".bash_history", {st_mode=S_IFREG|0600, st_size=2295, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr(".bash_history", "security.selinux", 0x55a0aef5ad50, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ad50 of the extended attribute identified by "security.selinux" and associated with ".bash_history" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr(".bash_history", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with ".bash_history" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

lstat("Images", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
## obtain information about the file pointed to by path "Images", {st_mode=S_IFDIR|0755, st_size=4096, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

lgetxattr("Images", "security.selinux", 0x55a0aef5ad70, 255) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value 0x55a0aef5ad70 of the extended attribute identified by "security.selinux" and associated with "Images" in the file system. The length of the attribute value is 255. Return -1, error data couldn't be found.

getxattr("Images", "system.posix_acl_access", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
## retrieves the value NULL of the extended attribute identified by "system.posix_acl_access" and associated with "Images" in the file system. The length of the attribute value is 0. Return -1, error data couldn't be found.

```
getxattr("Images", "system.posix_acl_default", NULL, 0) = -1 ENODATA (Aucune donnée disponible)
```

## retrieves the value NULL of the extended attribute identified by "system.posix_acl_default" and associated with "Images" in the file system. The length of the attribute value is 0. Return -1, error data couldn’t be found.

```
getdents64(3, /* 0 entries */, 32768)   = 0
```

## gets directory entries on file descriptor 3 into buffer pointed to by /* 0 entries */. 32768 is the size of this buffer. On end of directory, 0 is returned.

```
close(3)                                = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
openat(AT_FDCWD, "/usr/share/locale/fr_FR.UTF-8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr_FR.UTF-8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

```
openat(AT_FDCWD, "/usr/share/locale/fr_FR.utf8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr_FR.utf8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

```
openat(AT_FDCWD, "/usr/share/locale/fr_FR/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr_FR/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

```
openat(AT_FDCWD, "/usr/share/locale/fr.UTF-8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr.UTF-8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

```
openat(AT_FDCWD, "/usr/share/locale/fr.utf8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path

"/usr/share/locale/fr.utf8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale/fr/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/fr/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale-langpack/fr_FR.UTF-8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-langpack/fr_FR.UTF-8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale-langpack/fr_FR.utf8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-langpack/fr_FR.utf8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale-langpack/fr_FR/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-langpack/fr_FR/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale-langpack/fr.UTF-8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-langpack/fr.UTF-8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale-langpack/fr.utf8/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-

langpack/fr.utf8/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn't exist.

```
openat(AT_FDCWD, "/usr/share/locale-langpack/fr/LC_MESSAGES/coreutils.mo",
O_RDONLY) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-langpack/fr/LC_MESSAGES/coreutils.mo" to the flag O_RDONLY means only for reading. Returning 3 is the file descriptor (and it was a success).

```
fstat(3, {st_mode=S_IFREG|0644, st_size=383854, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=383854, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 383854, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb74ab9d000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 383854 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                          = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
```

## obtain information about the file descriptor 1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
write(1, "total 100\n", 10total 100)            = 10
```

## writes up to 10 bytes "total 100\n" to the file descriptor 1. The return value is the number of bytes.

```
openat(AT_FDCWD, "/etc/localtime", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/localtime" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

```
fstat(3, {st_mode=S_IFREG|0644, st_size=2962, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=2962, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
fstat(3, {st_mode=S_IFREG|0644, st_size=2962, ...}) = 0
```

## obtain information about the file descriptor 3,{st_mode=S_IFREG|0644, st_size=2962, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
read(3, "TZif2\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\r\0\0\0\r\0\0\0\0"..., 4096) = 2962
```

## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "TZif2\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\r\0\0\0\r\0\0\0\0". A smaller number of bytes is returned, so it is still a success.

```
lseek(3, -1863, SEEK_CUR)              = 1099
```

## used to change the location of the read/write pointer of a fd. 3 is the current fd, -1863 is the offset and SEEK_CUR means the offset is set to -1863 bytes. Return the resulting location of the offset.

```
read(3, "TZif2\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\r\0\0\0\r\0\0\0\0"..., 4096) = 1863
```

## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "TZif2\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\r\0\0\0\r\0\0\0\0". A smaller number of bytes is returned, so it is still a success.

```
close(3)                               = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
clock_gettime(CLOCK_REALTIME, {tv_sec=1636482627, tv_nsec=553247734}) = 0
```

```
write(1, "drwxr-xr-x 19 zheng zheng 4096 s"..., 48drwxr-xr-x 19 zheng zheng 4096 sept. 27 21:44 .) = 48
write(1, "drwxr-xr-x  3 root  root  4096 s"..., 49drwxr-xr-x  3 root  root  4096 sept. 26 15:28 ..) = 49
write(1, "-rw-------  1 zheng zheng 2295 o"..., 60-rw-------  1 zheng zheng 2295 oct.  21 20:23 .bash_history) = 60
write(1, "-rw-r--r--  1 zheng zheng  220 s"..., 59-rw-r--r--  1 zheng zheng  220 sept. 26 15:28 .bash_logout) = 59
write(1, "-rw-r--r--  1 zheng zheng 3771 s"..., 54-rw-r--r--  1 zheng zheng 3771 sept. 26 15:28 .bashrc) = 54
write(1, "drwxr-xr-x  2 zheng zheng 4096 s"..., 53drwxr-xr-x  2 zheng zheng 4096 sept. 27 20:30 Bureau) = 53
write(1, "drwxr-xr-x 15 zheng zheng 4096 o"..., 53drwxr-xr-x 15 zheng zheng 4096 oct.  21 13:31 .cache) = 53
write(1, "drwxr-xr-x 15 zheng zheng 4096 s"..., 54drwxr-xr-x 15 zheng zheng 4096 sept. 27 03:46 .config) = 54
write(1, "drwxr-xr-x  2 zheng zheng 4096 s"..., 56drwxr-xr-x  2 zheng zheng 4096 sept. 26 18:38 Documents) = 56
write(1, "drwx------  3 zheng zheng 4096 s"..., 53drwx------  3 zheng zheng 4096 sept. 27 19:45 .gnupg) = 53
write(1, "drwxr-xr-x  2 zheng zheng 4096 o"..., 53drwxr-xr-x  2 zheng zheng 4096 oct.  21
15:09 Images) = 53
write(1, "drwxr-xr-x  3 zheng zheng 4096 s"..., 53drwxr-xr-x  3 zheng zheng 4096 sept. 26
18:37 .local) = 53
write(1, "drwxr-xr-x  2 zheng zheng 4096 s"..., 55drwxr-xr-x  2 zheng zheng 4096 sept. 26
18:38 Modèles) = 55
write(1, "drwx------  5 zheng zheng 4096 s"..., 55drwx------  5 zheng zheng 4096 sept. 27
03:01 .mozilla) = 55
write(1, "drwxr-xr-x  2 zheng zheng 4096 s"..., 54drwxr-xr-x  2 zheng zheng 4096 sept. 26
18:38 Musique) = 54
write(1, "drwx------  3 zheng zheng 4096 s"..., 51drwx------  3 zheng zheng 4096 sept. 27
03:19 .pki) = 51
write(1, "-rw-r--r--  1 zheng zheng  807 s"..., 55-rw-r--r--  1 zheng zheng  807 sept. 26
15:28 .profile) = 55
write(1, "drwxr-xr-x  2 zheng zheng 4096 s"..., 53drwxr-xr-x  2 zheng zheng 4096 sept. 26
18:38 Public) = 53
write(1, "-rw-------  1 zheng zheng  256 s"..., 60-rw-------  1 zheng zheng  256 sept. 27
03:17 .pulse-cookie) = 60
write(1, "drwx------  3 zheng zheng 4096 s"..., 51drwx------  3 zheng zheng 4096 sept. 27
03:04 snap) = 51
write(1, "drwx------  2 zheng zheng 4096 s"..., 51drwx------  2 zheng zheng 4096 sept. 26
19:06 .ssh) = 51
write(1, "drwxrwxr-x  2 zheng zheng 4096 s"..., 53drwxrwxr-x  2 zheng zheng 4096 sept. 28
05:06 .steam) = 53
write(1, "lrwxrwxrwx  1 zheng zheng   30 s"..., 91lrwxrwxrwx  1 zheng zheng   30 sept. 27
21:44 .steampath -> /home/zheng/.steam/sdk32/steam) = 91
write(1, "lrwxrwxrwx  1 zheng zheng   28 s"..., 88lrwxrwxrwx  1 zheng zheng   28 sept. 27
21:44 .steampid -> /home/zheng/.steam/steam.pid) = 88
write(1, "-rw-r--r--  1 zheng zheng    0 s"..., 72-rw-r--r--  1 zheng zheng    0 sept. 27
03:10 .sudo_as_admin_successful) = 72
write(1, "drwxr-xr-x  4 zheng zheng 4096 o"..., 64drwxr-xr-x  4 zheng zheng 4096 oct.  21
13:32 Téléchargements) = 64
write(1, "drwxr-xr-x  2 zheng zheng 4096 s"..., 54drwxr-xr-x  2 zheng zheng 4096 sept. 26
18:38 Vidéos) = 54
write(1, "-rw-rw-r--  1 zheng zheng  177 s"..., 57-rw-rw-r--  1 zheng zheng  177 sept. 27
03:37 .wget-hsts) = 57
```

## SAME SYSTEM CALLS WITH DIFFERENT BUFFERS AND NUMBER OF BYTES

## writes up to *N* bytes "*differen*t" to the file descriptor 1. The return value is the number of bytes.

close(1)   = 0

## close the file descriptor 1. Return 0 means it was a success.

close(2)   = 0

## close the file descriptor 2. Return 0 means it was a success.

exit_group(0)   = ?

## exit all threads in the calling process. End of the strace.
+++ exited with 0 +++

---

strace pwd

execve("/usr/bin/pwd", ["pwd"], 0x7ffd6e069c70 /* 48 vars */) = 0
## used for executing a program; using as arguments the path ("/usr/bin/pwd"), the command that we want (["pwd") and the environment of the program (0x7ffd6e069c70). Returning 0 means it was a success.

brk(NULL)   = 0x560d3be68000
## used to make a break in the process; the value 0x560d3be68000 means that the allocated memory for the next address is going to be resume

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe94935430) = -1 EINVAL (Argument invalide)
## set an architecture process or a thread state. 0x3001 is the selected subfunction and 0x7ffe94935430 its address. Returning -1 EINVAL means that the selected subfunction is not valid.

access("/etc/ld.so.preload", R_OK)   = -1 ENOENT (Aucun fichier ou dossier de ce type)
## check if the calling process has access to this filename path "/etc/ld.so.preload", R_OK means that it is trying to read. Returning -1 ENOENT means an error and that such a file doesn't exist.

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/ld.so.cache" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=80743, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=80743, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 80743, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1cc0cb4000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 80743 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents

others from using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path " "/lib/x86_64-linux-gnu/libc.so.6" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

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
## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "\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". The number of bytes is returned, so it is a success.

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
## SAME SYSTEM CALL WITH DIFFERENT PATHS, BYTE SIZES AND OFFSET
## attempts to read up to *N* bytes from file descriptor 3 at *offset* into the buffer starting at "*different*". Number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=2029224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1cc0cb2000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

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
```

## SAME SYSTEM CALL WITH DIFFERENT PATHS, BYTE SIZES AND OFFSET

## attempts to read up to *N* bytes from file descriptor 3 at *offset* into the buffer starting at "*different*". Number of bytes is returned, so it was a success.

```
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f1cc0ac0000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 2036952 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
mprotect(0x7f1cc0ae5000, 1847296, PROT_NONE) = 0
```

## changes the access protections for the calling process's memory. 0x7f1cc0ae5000 is the address of a region in the memory, 1847296 is the size of the protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

```
mmap(0x7f1cc0ae5000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f1cc0ae5000
mmap(0x7f1cc0c5d000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f1cc0c5d000
mmap(0x7f1cc0ca8000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f1cc0ca8000
mmap(0x7f1cc0cae000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f1cc0cae000
```

## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES

## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag and MAP_ANONYMOUS means all the value are resetting to 0, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                                = 0
```

## close the file descriptor 3. Return 0 means it was a success.

arch_prctl(ARCH_SET_FS, 0x7f1cc0cb3580) = 0
## set an architecture process or a thread state. ARCH_SET_FS set the 64bit base for the FS register to 0x7f1cc0cb3580 address. Returning 0 means it was a success.

mprotect(0x7f1cc0ca8000, 12288, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7f1cc0ca8000 is the address of a region in the memory, 12288 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x560d3a126000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x560d3a126000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7f1cc0cf5000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7f1cc0cf5000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

munmap(0x7f1cc0cb4000, 80743)   = 0
## deletes the address for the specified address region, and unmapps it. 0x7f1cc0cb4000 is the specified address region and 80743 its length (not necessarily). Return 0 meaning it was a success.

brk(NULL)   = 0x560d3be68000
## used to make a break in the process; the value 0x560d3be68000 means that the allocated memory for the next address is going to be resumed.

brk(0x560d3be89000)   = 0x560d3be89000
## used to make a break in the process; the value 0x560d3be89000 means that the allocated memory for the next address is going to be resumed.

openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/lib/locale/locale-archive" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=14537584, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=14537584, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 14537584, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1cbfce2000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 14537584 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

getcwd("/home/zheng", 4096)   = 12
## copies an absolute pathname of the current working directory to the array pointed to by "/home/zheng", which is of 4096 size. Return a pointer to the buffer path.

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
## obtain information about the file descriptor 1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0),...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

write(1, "/home/zheng\n", 12/home/zheng)   = 12
## writes up to 12 bytes "/home/zheng\n" to the file descriptor 1. The return value is the number of bytes.

close(1)   = 0
## close the file descriptor 1. Return 0 means it was a success.
close(2)   = 0
## close the file descriptor 2. Return 0 means it was a success.

exit_group(0)   = ?
## exit all threads in the calling process. End of the strace.

+++ exited with 0 +++

---

strace ping -c 1 8.8.8.8

execve("/usr/bin/ping", ["ping", "-c", "1", "8.8.8.8"], 0x7fffa2d02f48 /* 48 vars */) = 0
## used for executing a program; using as arguments the path ("/usr/bin/ping"), the command that we want (["ping", "-c", "1", "8.8.8.8"]) and the environment of the program (0x7fffa2d02f48). Returning 0 means it was a success.

access("/etc/suid-debug", F_OK)   = -1 ENOENT (Aucun fichier ou dossier de ce type)
## check if the calling process has access to this filename path "/etc/suid-debug", F_OK means that it is trying to find if such a file exists. Returning -1 ENOENT means an error and that such a file doesn't exist.

brk(NULL)   = 0x559df1665000
## used to make a break in the process; the value 0x559df1665000 means that the allocated memory for the next address is going to be resumed.

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd25cbf4e0) = -1 EINVAL (Argument invalide)
## set an architecture process or a thread state. 0x3001 is the selected subfunction and 0x7ffd25cbf4e0 its address. Returning -1 EINVAL means that the selected subfunction is not valid.

fcntl(0, F_GETFD)   = 0
fcntl(1, F_GETFD)   = 0
fcntl(2, F_GETFD)   = 0
## SAME SYSTEM CALL WITH DIFFERENT FD
return (as the function result) the file descriptor flags.

access("/etc/suid-debug", F_OK)   = -1 ENOENT (Aucun fichier ou dossier de ce type)
## check if the calling process has access to this filename path "/etc/suid-debug", F_OK means that it is trying to find if such a file exists. Returning -1 ENOENT means an error and that such a file doesn't exist.

access("/etc/ld.so.preload", R_OK)   = -1 ENOENT (Aucun fichier ou dossier de ce type)
## check if the calling process has access to this filename path "/etc/ld.so.preload", R_OK means that it is trying to read. Returning -1 ENOENT means an error and that such a file doesn't exist.

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path " "/etc/ld.so.cache" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=80743, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=80743, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 80743, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb9dbd2d000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 80743 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                    = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libcap.so.2", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libcap.so.2" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

```
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300#\0\0\0\0\0\0"..., 832) = 832
```

## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300#\0\0\0\0\0\0". The number of bytes is returned, so it is a success.

```
fstat(3, {st_mode=S_IFREG|0644, st_size=31120, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=31120, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fb9dbd2b000
mmap(NULL, 33112, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb9dbd22000
```

## SAME SYSTEM CALL WITH DIFFERENT LENGTH

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag and MAP_ANONYMOUS means all the value are resetting to 0, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
mprotect(0x7fb9dbd24000, 20480, PROT_NONE) = 0
```

## changes the access protections for the calling process's memory. 0x7fb9dbd24000 is the address of a region in the memory, 20480 is the size of the

protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

mmap(0x7fb9dbd24000, 12288, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fb9dbd24000
mmap(0x7fb9dbd27000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x5000) = 0x7fb9dbd27000
mmap(0x7fb9dbd29000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7fb9dbd29000
## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES
## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcrypt.so.20", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libgcrypt.so.20" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200\305\0\0\0\0\0\0"..., 832) = 832
## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200\305\0\0\0\0\0\0". The number of bytes is returned, so it is a success.

fstat(3, {st_mode=S_IFREG|0644, st_size=1168056, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=1168056, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 1171400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fb9dbc04000
mmap(0x7fb9dbc10000, 843776, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xc000) = 0x7fb9dbc10000
mmap(0x7fb9dbcde000, 249856, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xda000) = 0x7fb9dbcde000
mmap(0x7fb9dbd1b000, 28672, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x116000) = 0x7fb9dbd1b000

## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES
## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libresolv.so.2", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libresolv.so.2" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 G\0\0\0\0\0\0"..., 832) = 832
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 G\0\0\0\0\0\0". The number of bytes is returned, so it is a success.

fstat(3, {st_mode=S_IFREG|0644, st_size=101320, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=101320, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 113280, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb9dbbe8000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 113280 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE|MAP_DENYWRITE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mprotect(0x7fb9dbbec000, 81920, PROT_NONE) = 0
## changes the access protections for the calling process's memory. 0x7fb9dbbec000 is the address of a region in the memory, 81920 is the size of the protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

```
mmap(0x7fb9dbbec000, 65536, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7fb9dbbec000
mmap(0x7fb9dbbfc000, 12288, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14000) = 0x7fb9dbbfc000
mmap(0x7fb9dbc00000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17000) = 0x7fb9dbc00000
mmap(0x7fb9dbc02000, 6784, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb9dbc02000
```

## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES

## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag and MAP_ANONYMOUS means all the value are resetting to 0, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                    = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libc.so.6" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

```
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
```

## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "\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". The number of bytes is returned, so it is a success.

```
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
```

## SAME SYSTEM CALL WITH DIFFERENT PATHS, BYTE SIZES AND OFFSET

## attempts to read up to *N* bytes from file descriptor 3 at *offset* into the buffer starting at "*different*". Number of bytes is returned, so it was a success.

```
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=2029224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
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
```

## SAME SYSTEM CALL WITH DIFFERENT PATHS, BYTE SIZES AND OFFSET

## attempts to read up to *N* bytes from file descriptor 3 at *offset* into the buffer starting at "*different*". Number of bytes is returned, so it was a success.

mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb9db9f6000

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 2036952 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE|MAP_DENYWRITE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mprotect(0x7fb9dba1b000, 1847296, PROT_NONE) = 0

## changes the access protections for the calling process's memory. 0x7fb9dba1b000 is the address of a region in the memory, 1847296 is the size of the protection, PROT_NONE means the memory cannot be accessed. Returning 0 means it was a success.

```
mmap(0x7fb9dba1b000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7fb9dba1b000
mmap(0x7fb9dbb93000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7fb9dbb93000
mmap(0x7fb9dbbde000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fb9dbbde000
mmap(0x7fb9dbbe4000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb9dbbe4000
```

## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES

## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag and MAP_ANONYMOUS means all the value are resetting to 0, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0

## close the file descriptor 3. Return 0 means it was a success.

```
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgpg-error.so.0",
O_RDONLY|O_CLOEXEC) = 3
```

## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libgpg-error.so.0" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

```
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`L\0\0\0\0\0\0"..., 832) = 832
```

## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`L\0\0\0\0\0\0. The number of bytes is returned, so it is a success.

```
fstat(3, {st_mode=S_IFREG|0644, st_size=137584, ...}) = 0
```

## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=137584, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
mmap(NULL, 139872, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fb9db9d3000
mmap(0x7fb9db9d7000, 77824, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7fb9db9d7000
mmap(0x7fb9db9ea000, 40960, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17000) = 0x7fb9db9ea000
mmap(0x7fb9db9f4000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x20000) = 0x7fb9db9f4000
```

## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES

## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                    = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fb9db9d1000
```

## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE mens that the mapping is going to private (prevents other for using it) and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

arch_prctl(ARCH_SET_FS, 0x7fb9db9d2040) = 0
## set an architecture process or a thread state. ARCH_SET_FS set the 64bit base for the FS register to 0x7fb9db9d2040 address. Returning 0 means it was a success.

mprotect(0x7fb9dbbde000, 12288, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb9dbbde000 is the address of a region in the memory, 12288 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb9db9f4000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb9db9f4000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb9dbc00000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb9dbc00000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb9dbd1b000, 8192, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb9dbd1b000 is the address of a region in the memory, 8192 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb9dbd29000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb9dbd29000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x559def890000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x559def890000 is the address of a region in the memory, 4096 is the size of the protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

mprotect(0x7fb9dbd6e000, 4096, PROT_READ) = 0
## changes the access protections for the calling process's memory. 0x7fb9dbd6e000 is the address of a region in the memory, 4096 is the size of the

protection, PROT_READ means the memory can be read. Returning 0 means it was a success.

```
munmap(0x7fb9dbd2d000, 80743)          = 0
```
## deletes the address for the specified address region, and unmapps it. 0x7fb9dbd2d000 is the specified address region and 80743 its length (not necessarily). Return 0 meaning it was a success.

```
brk(NULL)                    = 0x559df1665000
```
## used to make a break in the process; the value 0x559df1665000 means that the allocated memory for the next address is going to be resumed.

```
brk(0x559df1686000)                 = 0x559df1686000
```
## used to make a break in the process; the value 0x559df1686000 means that the allocated memory for the next address is going to be resumed.

```
prctl(PR_CAPBSET_READ, CAP_MAC_OVERRIDE) = 1
prctl(PR_CAPBSET_READ, 0x30 /* CAP_??? */) = -1 EINVAL (Argument invalide)
prctl(PR_CAPBSET_READ, 0x28 /* CAP_??? */) = 1
prctl(PR_CAPBSET_READ, 0x2c /* CAP_??? */) = -1 EINVAL (Argument invalide)
prctl(PR_CAPBSET_READ, 0x2a /* CAP_??? */) = -1 EINVAL (Argument invalide)
prctl(PR_CAPBSET_READ, 0x29 /* CAP_??? */) = -1 EINVAL (Argument invalide)
```
## SAME SYSTEM CALLS WITH DIFFERENT ARGUMENTS
## manipulates various aspects of the behavior of the calling thread or process. Return 1 if the capability specified in *argument* is in the calling thread's capability bounding set. Return -1 in case of an error.

```
capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, NULL) = 0
capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, {effective=0, permitted=0,
inheritable=0}) = 0
capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, NULL) = 0
capset({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, {effective=0, permitted=0,
inheritable=0}) = 0
```
## SAME SYSTEM CALLS
## set/get capabilities of thread(s). Return 0 on success.

```
prctl(PR_SET_KEEPCAPS, 1)             = 0
```
## manipulates various aspects of the behavior of the calling thread or process. Set the state of the calling thread’s flag. If 0 or 1, the flag is going to reset on 0. Return 0 on success.

```
getuid()                       = 1000
```
## returns the real user ID of the calling process.

setuid(1000) = 0
## sets the effective user ID of the calling process.

prctl(PR_SET_KEEPCAPS, 0) = 0
## manipulates various aspects of the behavior of the calling thread or process. Set the state of the calling thread's flag. If 0 or 1, the flag is going to reset on 0. Return 0 on success.

getuid() = 1000
## returns the real user ID of the calling process.

geteuid() = 1000
## returns the effective user ID of the calling process.

openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/lib/locale/locale-archive" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=14537584, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=14537584, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 14537584, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb9dabf3000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 14537584 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE mens that the mapping is going to private (prevents other for using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3) = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale/locale.alias" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=2996, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=2996, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

read(3, "# Locale name alias data base.\n#"..., 4096) = 2996
## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "# Locale name alias data base.\n#". A smaller number of bytes is returned, so it is still a success.

read(3, "", 4096)   = 0
## attempts to read up to 4096 bytes from file descriptor 3 into the buffer starting at "". Return 0 on success meaning it is the end of the file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

```
openat(AT_FDCWD, "/usr/share/locale/fr_FR.UTF-8/LC_MESSAGES/iputils.mo",
O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale/fr_FR.utf8/LC_MESSAGES/iputils.mo", O_RDONLY)
= -1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale/fr_FR/LC_MESSAGES/iputils.mo", O_RDONLY) = -1
ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale/fr.UTF-8/LC_MESSAGES/iputils.mo", O_RDONLY) =
-1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale/fr.utf8/LC_MESSAGES/iputils.mo", O_RDONLY) = -1
ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale/fr/LC_MESSAGES/iputils.mo", O_RDONLY) = -1
ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale-langpack/fr_FR.UTF-8/LC_MESSAGES/iputils.mo",
O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale-langpack/fr_FR.utf8/LC_MESSAGES/iputils.mo",
O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale-langpack/fr_FR/LC_MESSAGES/iputils.mo",
O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale-langpack/fr.UTF-8/LC_MESSAGES/iputils.mo",
O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
openat(AT_FDCWD, "/usr/share/locale-langpack/fr.utf8/LC_MESSAGES/iputils.mo",
O_RDONLY) = -1 ENOENT (Aucun fichier ou dossier de ce type)
```

## SAME SYSTEM CALL WITH DIFFERENT PATHS
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "different" to the flag O_RDONLY means only for reading. Returning -1 is an error meaning such a file doesn’t exist.

openat(AT_FDCWD, "/usr/share/locale-langpack/fr/LC_MESSAGES/iputils.mo", O_RDONLY) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/share/locale-langpack/fr/LC_MESSAGES/iputils.mo" to the flag O_RDONLY means only for reading. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=13277, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=13277, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 13277, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb9dbd3d000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 13277 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE mens that the mapping is going to private (prevents other for using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, NULL) = 0
capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, {effective=0, permitted=0, inheritable=0}) = 0
## SAME SYSTEM CALLS
## set/get capabilities of thread(s). Return 0 on success.

socket(AF_INET, SOCK_DGRAM, IPPROTO_ICMP) = 3
## creates an endpoint for communication. The communication domain is AF_INET, SOCK_DGRAM supports datagrams. Return the new fd.

socket(AF_INET6, SOCK_DGRAM, IPPROTO_ICMPV6) = 4
## creates an endpoint for communication. The communication domain is AF_INET6, SOCK_DGRAM supports datagrams. Return the new fd.

capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, NULL) = 0
capget({version=_LINUX_CAPABILITY_VERSION_3, pid=0}, {effective=0, permitted=0, inheritable=0}) = 0
## SAME SYSTEM CALLS
## set/get capabilities of thread(s). Return 0 on success.

socket(AF_INET, SOCK_DGRAM, IPPROTO_IP) = 5
## creates an endpoint for communication. The communication domain is AF_INET, SOCK_DGRAM supports datagrams. Return the new fd.

connect(5, {sa_family=AF_INET, sin_port=htons(1025), sin_addr=inet_addr("8.8.8.8")}, 16) = 0
## connects the socket referred to by the file descriptor 5 to the address {sa_family=AF_INET, sin_port=htons(1025), sin_addr=inet_addr("8.8.8.8")} with a size of 16. Return 0 on success.

getsockname(5, {sa_family=AF_INET, sin_port=htons(47741), sin_addr=inet_addr("10.42.4.134")}, [16]) = 0
## manipulate options for the socket referred to by the file descriptor 5, in the buffer pointed to by {sa_family=AF_INET, sin_port=htons(47741), sin_addr=inet_addr("10.42.4.134")} address. 16 is the size in bytes of the address. Return 0 on success.

close(5)   = 0
## close the file descriptor 5. Return 0 means it was a success.

setsockopt(3, SOL_IP, IP_RECVERR, [1], 4) = 0

setsockopt(3, SOL_IP, IP_RECVTTL, [1], 4) = 0

setsockopt(3, SOL_IP, IP_RETOPTS, [1], 4) = 0

setsockopt(3, SOL_SOCKET, SO_SNDBUF, [324], 4) = 0

setsockopt(3, SOL_SOCKET, SO_RCVBUF, [65536], 4) = 0

getsockopt(3, SOL_SOCKET, SO_RCVBUF, [131072], [4]) = 0

## SAME SYSTEM CALLS WITH DIFFERENT OPTIONS AND SIZES
## manipulate options for the socket referred to by the file descriptor 3, *option* is not interpreted, *size* is the size in bytes of the buffer. Return 0 on success.

openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache", O_RDONLY) = 5
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache" to the flag O_RDONLY means only for reading. Returning 5 is the file descriptor (and it was a success).

fstat(5, {st_mode=S_IFREG|0644, st_size=27002, ...}) = 0
## obtain information about the file descriptor 5, {st_mode=S_IFREG|0644, st_size=27002, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 27002, PROT_READ, MAP_SHARED, 5, 0) = 0x7fb9dbd36000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 27002 is the length

of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_SHARED shares this mapping, 5 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(5)   = 0
## close the file descriptor 5. Return 0 means it was a success.

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
## obtain information about the file descriptor 1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

write(1, "PING 8.8.8.8 (8.8.8.8) 56(84) by"..., 45PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.) = 45
## writes up to 45 bytes "PING 8.8.8.8 (8.8.8.8) 56(84) by" to the file descriptor 1. The return value is the number of bytes.

setsockopt(3, SOL_SOCKET, SO_TIMESTAMP_OLD, [1], 4) = 0
setsockopt(3, SOL_SOCKET, SO_SNDTIMEO_OLD, "\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 16) = 0
setsockopt(3, SOL_SOCKET, SO_RCVTIMEO_OLD, "\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 16) = 0
## SAME SYSTEM CALLS WITH DIFFERENT OPTIONS AND SIZES
## manipulate options for the socket referred to by the file descriptor 3, *option* is not interpreted, *size* is the size in bytes of the buffer. Return 0 on success.

rt_sigaction(SIGINT, {sa_handler=0x559def886740, sa_mask=[], sa_flags=SA_RESTORER, sa_restorer=0x7fb9dba3c210}, NULL, 8) = 0
used to change the action taken by a process on receipt of a specific signal. This new action {sa_handler=0x559def886740, sa_mask=[], sa_flags=SA_RESTORER, sa_restorer=0x7fb9dba3c210} for signal SIGINT is saved. 8 is the size in bytes for sa_mask. Return 0 on success.

rt_sigaction(SIGALRM, {sa_handler=0x559def886740, sa_mask=[], sa_flags=SA_RESTORER, sa_restorer=0x7fb9dba3c210}, NULL, 8) = 0
used to change the action taken by a process on receipt of a specific signal. This new action {sa_handler=0x559def886740, sa_mask=[], sa_flags=SA_RESTORER, sa_restorer=0x7fb9dba3c210} for signal SIGALRM is saved. 8 is the size in bytes for sa_mask. Return 0 on success.

rt_sigaction(SIGQUIT, {sa_handler=0x559def886730, sa_mask=[], sa_flags=SA_RESTORER, sa_restorer=0x7fb9dba3c210}, NULL, 8) = 0
used to change the action taken by a process on receipt of a specific signal. This new action {sa_handler=0x559def886730, sa_mask=[],

sa_flags=SA_RESTORER, sa_restorer=0x7fb9dba3c210} for signal SIGQUIT is saved. 8 is the size in bytes for sa_mask. Return 0 on success.

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
## used to fetch and/or change the signal mask of the calling thread. Since SIG_SETMASK then the set of blocked signals is set to []. 8 is the size in bytes for set []. Return 0 on success.

gettimeofday({tv_sec=1636482773, tv_usec=221341}, NULL) = 0
## get the time of the day. Return 0 on success.

ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
## manipulates the underlying device parameters of special files. 1 is the file descriptor, TCGETS gets the current serial port settings. Return 0 on success.

ioctl(1, TIOCGWINSZ, {ws_row=24, ws_col=80, ws_xpixel=0, ws_ypixel=0}) = 0
## manipulates the underlying device parameters of special files. 1 is the file descriptor, TIOCGWINSZ gets window size. Return 0 on success.

gettimeofday({tv_sec=1636482773, tv_usec=221529}, NULL) = 0
gettimeofday({tv_sec=1636482773, tv_usec=221590}, NULL) = 0
## SAME SYSTEM CALLS WITH DIFFERENT VALUE OF TIMEZONE
## get the time of the day. Return 0 on success.

sendto(3, "\10\0?\252\0\0\0\1\325\276\212a\0\0\0\0\226a\3\0\0\0\0\0\20\21\22\23\24\25\26\27"..., 64, 0, {sa_family=AF_INET, sin_port=htons(0), sin_addr=inet_addr("8.8.8.8")}, 16) = 64
## send a message on socket 3, the message is "\10\0?\252\0\0\0\1\325\276\212a\0\0\0\0\226a\3\0\0\0\0\0\20\21\22\23\24\25\26\27" in a size of 64 bytes. Return the number of bytes sent.

setitimer(ITIMER_REAL, {it_interval={tv_sec=0, tv_usec=0}, it_value={tv_sec=10, tv_usec=0}}, NULL) = 0
## set the value of an interval timer. Return 0 on success.

recvmsg(3, {msg_name={sa_family=AF_INET, sin_port=htons(0), sin_addr=inet_addr("8.8.8.8")}, msg_namelen=128->16, msg_iov=[{iov_base="\0\0G\251\0\1\0\1\325\276\212a\0\0\0\0\226a\3\0\0\0\0\0\20\21\22\23\24\25\26\27"..., iov_len=192}], msg_iovlen=1, msg_control=[{cmsg_len=32, cmsg_level=SOL_SOCKET, cmsg_type=SO_TIMESTAMP_OLD, cmsg_data={tv_sec=1636482773, tv_usec=225454}}, {cmsg_len=20, cmsg_level=SOL_IP, cmsg_type=IP_TTL, cmsg_data=[119]}], msg_controllen=56, msg_flags=0}, 0) = 64
## received the message from socket 3. Return the number of bytes received.

write(1, "64\302\240octets de 8.8.8.8\302\240: icmp_se"..., 5964 octets de 8.8.8.8 : icmp_seq=1 ttl=119 temps=3.86 ms) = 59
write(1, "\n", 1)   = 1
write(1, "--- statistiques ping 8.8.8.8 --"..., 34--- statistiques ping 8.8.8.8 ---) = 34
write(1, "1\302\240paquets transmis, 1 re\303\247us, 0"..., 641 paquets transmis, 1 reçus, 0 % paquets perdus, temps 0 ms) = 64
write(1, "rtt min/avg/max/mdev = 3.864/3.8"..., 50rtt min/avg/max/mdev = 3.864/3.864/3.864/0.000 ms) = 50

## SAME SYSTEM CALLS WITH DIFFERENT BUFFERS AND NUMBER OF BYTES

writes up to *N* bytes "*differen*t" to the file descriptor 1. The return value is the number of bytes.

close(1)   = 0

## close the file descriptor 1. Return 0 means it was a success.

close(2)   = 0

## close the file descriptor 2. Return 0 means it was a success.

exit_group(0)   = ?

## exit all threads in the calling process. End of the strace.

+++ exited with 0 +++

---

strace "./hello"

execve("./hello", ["./hello"], 0x7ffde29e2d60 /* 49 vars */) = 0

## used for executing a program; using as arguments the path ("./hello"), the command that we want (["./hello"]) and the environment of the program (0x7ffde29e2d60). Returning 0 means it was a success.

brk(NULL)   = 0x564acf64f000

## used to make a break in the process; the value 0x564acf64f000 means that the allocated memory for the next address is going to be resumed.

arch_prctl(0x3001 /* ARCH_??? */, 0x7fff35d0b4f0) = -1 EINVAL (Argument invalide)

## set an architecture process or a thread state. 0x3001 is the selected subfunction and 0x7fff35d0b4f0 its address. Returning -1 EINVAL means that the selected subfunction is not valid.

access("/etc/ld.so.preload", R_OK)   = -1 ENOENT (Aucun fichier ou dossier de ce type)

## check if the calling process has access to this filename path "/etc/ld.so.preload", R_OK means that it is trying to read. Returning -1 ENOENT means an error and that such a file doesn’t exist.

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/etc/ld.so.cache" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

fstat(3, {st_mode=S_IFREG|0644, st_size=80743, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0644, st_size=80743, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 80743, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f76bb18c000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 80743 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

close(3)   = 0
## close the file descriptor 3. Return 0 means it was a success.

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
## creates or opens a file descriptor, AT_FDCWD is the pathname will be resolved relative to the current working directory, the file path "/lib/x86_64-linux-gnu/libc.so.6" to the flag O_RDONLY|O_CLOEXEC means only for reading and enabling the close-on-exec. Returning 3 is the file descriptor (and it was a success).

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
## attempts to read up to 832 bytes from file descriptor 3 into the buffer starting at "\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". The number of bytes is returned, so it is a success.

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
## SAME SYSTEM CALL WITH DIFFERENT PATHS, BYTE SIZES AND OFFSET
## attempts to read up to *N* bytes from file descriptor 3 at *offset* into the buffer starting at "*different*". Number of bytes is returned, so it was a success.

fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
## obtain information about the file descriptor 3, {st_mode=S_IFREG|0755, st_size=2029224, ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f76bb18a000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 8192 is the length of the mapping, PROT_READ|PROT_WRITE is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and MAP_ANONYMOUS means all the value are resetting to 0, -1 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

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

## SAME SYSTEM CALL WITH DIFFERENT PATHS, BYTE SIZES AND OFFSET
## attempts to read up to *N* bytes from file descriptor 3 at *offset* into the buffer starting at "*different*". Number of bytes is returned, so it was a success.

mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76baf98000
## creates a new map file to memory. NULL represents that no address was inputted for the map location (the kernel will directly assign an address), 2036952 is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it) and we can ignore the MAP_DENYWRITE flag, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

mprotect(0x7f76bafbd000, 1847296, PROT_NONE) = 0
## changes the access protections for the calling process's memory. 0x7f76bafbd000 is the address of a region in the memory, 1847296 is the size of the protection, PROT_NONE means the memory can be accessed. Returning 0 means it was a success.

mmap(0x7f76bafbd000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f76bafbd000

mmap(0x7f76bb135000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f76bb135000

```
mmap(0x7f76bb180000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f76bb180000
mmap(0x7f76bb186000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f76bb186000
```

## SAME SYSTEM CALL WITH DIFFERENT ADDRESSES

## creates a new map file to memory. *Address* represents that an address was inputted for the map location, *length* is the length of the mapping, PROT_READ is the protection of the memory and it means that we can read and write it, MAP_PRIVATE means that the mapping is going to private (prevents others from using it), we can ignore the MAP_DENYWRITE flag and MAP_ANONYMOUS means all the value are resetting to 0, 3 is the file descriptor and 0 means that there is no offset. The returned value is the address of the map file.

```
close(3)                    = 0
```

## close the file descriptor 3. Return 0 means it was a success.

```
arch_prctl(ARCH_SET_FS, 0x7f76bb18b540) = 0
```

```
mprotect(0x7f76bb180000, 12288, PROT_READ) = 0
mprotect(0x564acd929000, 4096, PROT_READ) = 0
mprotect(0x7f76bb1cd000, 4096, PROT_READ) = 0
```

## SAME SYSTEM CALLS WITH DIFFERENT ADDRESSES AND SIZES

## changes the access protections for the calling process's memory. *Adress* is the address of a region in the memory, *size* is the size of the protection, PROT_read means the memory can be read. Returning 0 means it was a success.

```
munmap(0x7f76bb18c000, 80743)           = 0
```

## deletes the address for the specified address region, and unmapps it. 0x7f76bb18c000 is the specified address region and 80743 its length (not necessarily). Return 0 meaning it was a success.

```
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
```

## obtain information about the file descriptor 1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...} means that if st_mode is valid, then the stat structure pointed can be updated. Returning 0 means it was a success.

```
brk(NULL)                        = 0x564acf64f000
brk(0x564acf670000)                   = 0x564acf670000
```

## SAME SYSTEM CALLS

## used to make a break in the process; the value 0x564acf670000 means that the allocated memory for the next address is going to be resumed

```
write(1, "Hello, World!\n", 14Hello, World!)        = 14
```

## writes up to 14 bytes "Hello World!\n" to the file descriptor 1. The return value is the number of bytes.

```
exit_group(0)                          = ?
## exit  all threads in the calling process. End of the strace.
+++ exited with 0 +++
```