Neculae Andrei-Fabian

Grupa 252

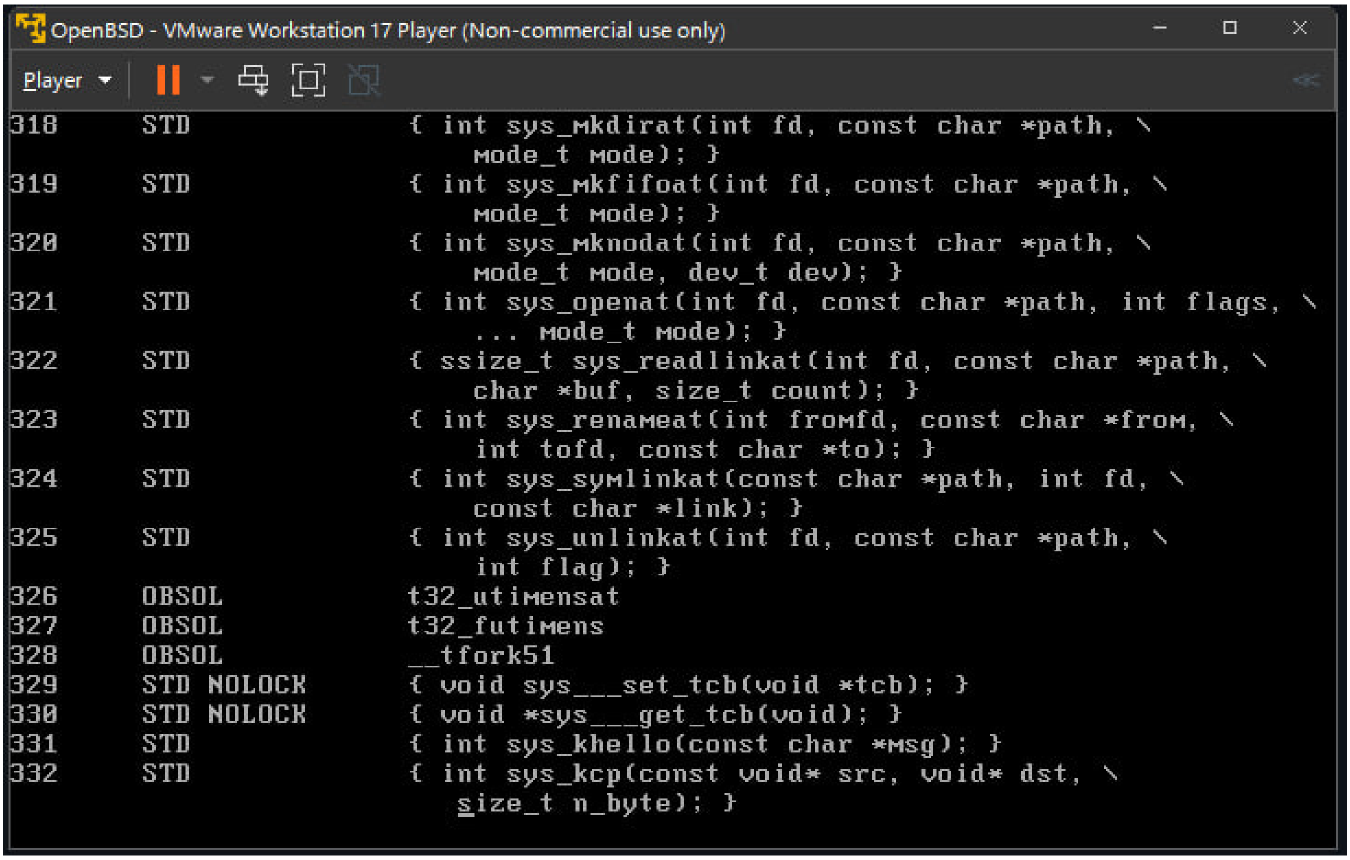
Sisteme de operare

Laboratorul 3

/sys/kern/syscalls.master

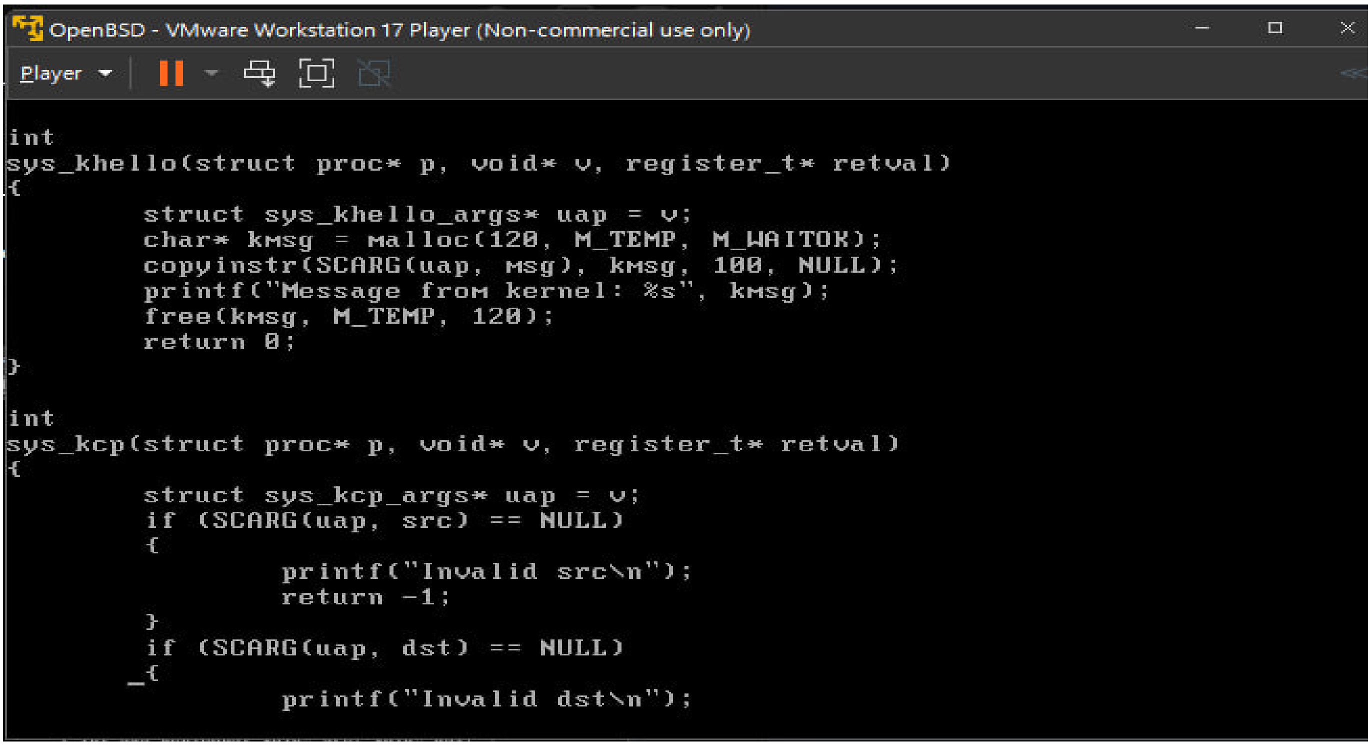
ID 331 = sys_khello

ID 332 = sys_kcp


OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

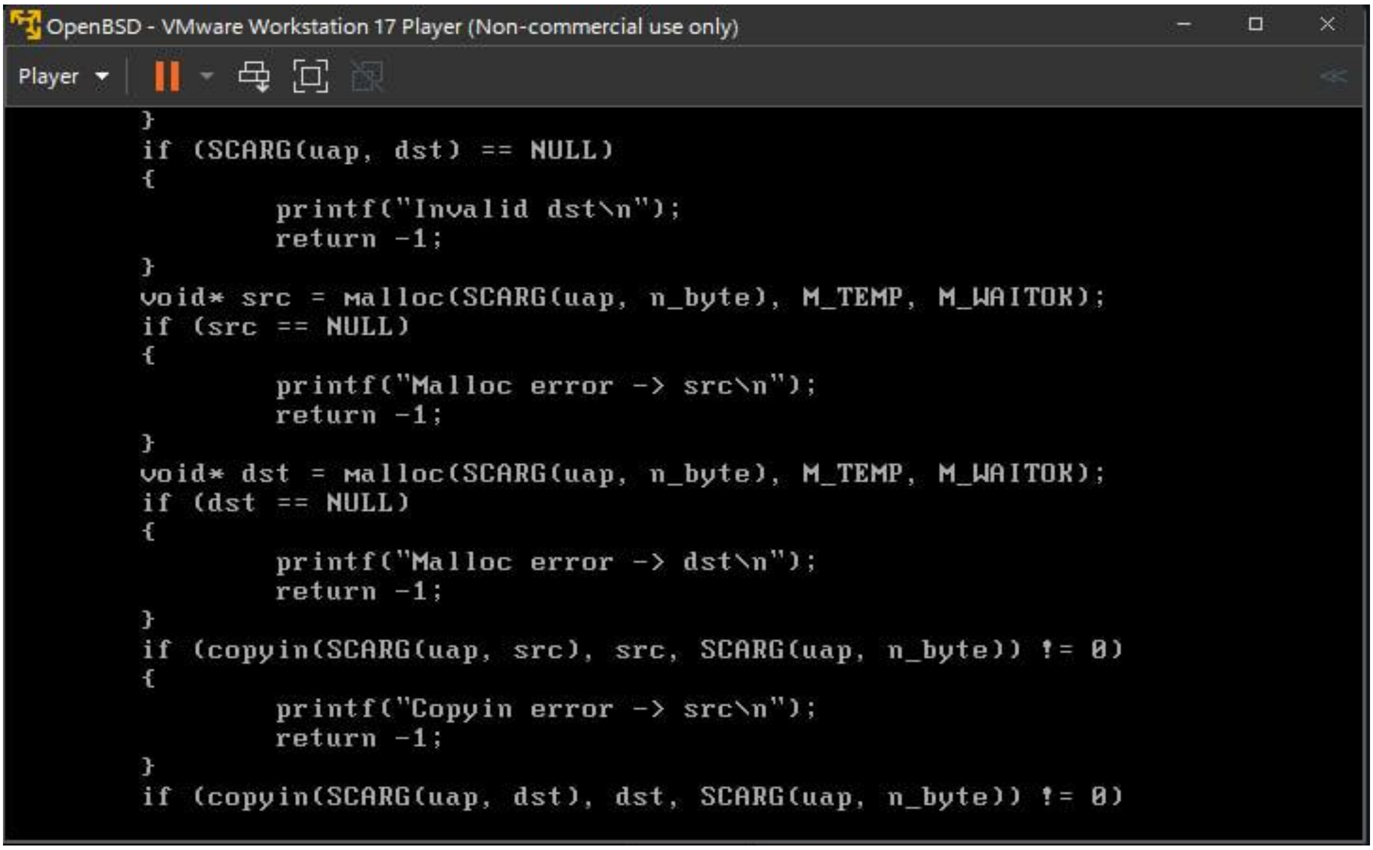
```
318     STD             { int sys_mkdirat(int fd, const char *path, \
                            mode_t mode); }
319     STD             { int sys_mkfifoat(int fd, const char *path, \
                            mode_t mode); }
320     STD             { int sys_mknodat(int fd, const char *path, \
                            mode_t mode, dev_t dev); }
321     STD             { int sys_openat(int fd, const char *path, int flags, \
                            ... mode_t mode); }
322     STD             { ssize_t sys_readlinkat(int fd, const char *path, \
                            char *buf, size_t count); }
323     STD             { int sys_renameat(int fromfd, const char *from, \
                            int tofd, const char *to); }
324     STD             { int sys_symlinkat(const char *path, int fd, \
                            const char *link); }
325     STD             { int sys_unlinkat(int fd, const char *path, \
                            int flag); }
326     OBSOL           t32_utimensat
327     OBSOL           t32_futimens
328     OBSOL           __tfork51
329     STD NOLOCK      { void sys___set_tcb(void *tcb); }
330     STD NOLOCK      { void *sys___get_tcb(void); }
331     STD             { int sys_khello(const char *msg); }
332     STD             { int sys_kcp(const void* src, void* dst, \
                           size_t n_byte); }
```

/sys/kern/sys_generic.c

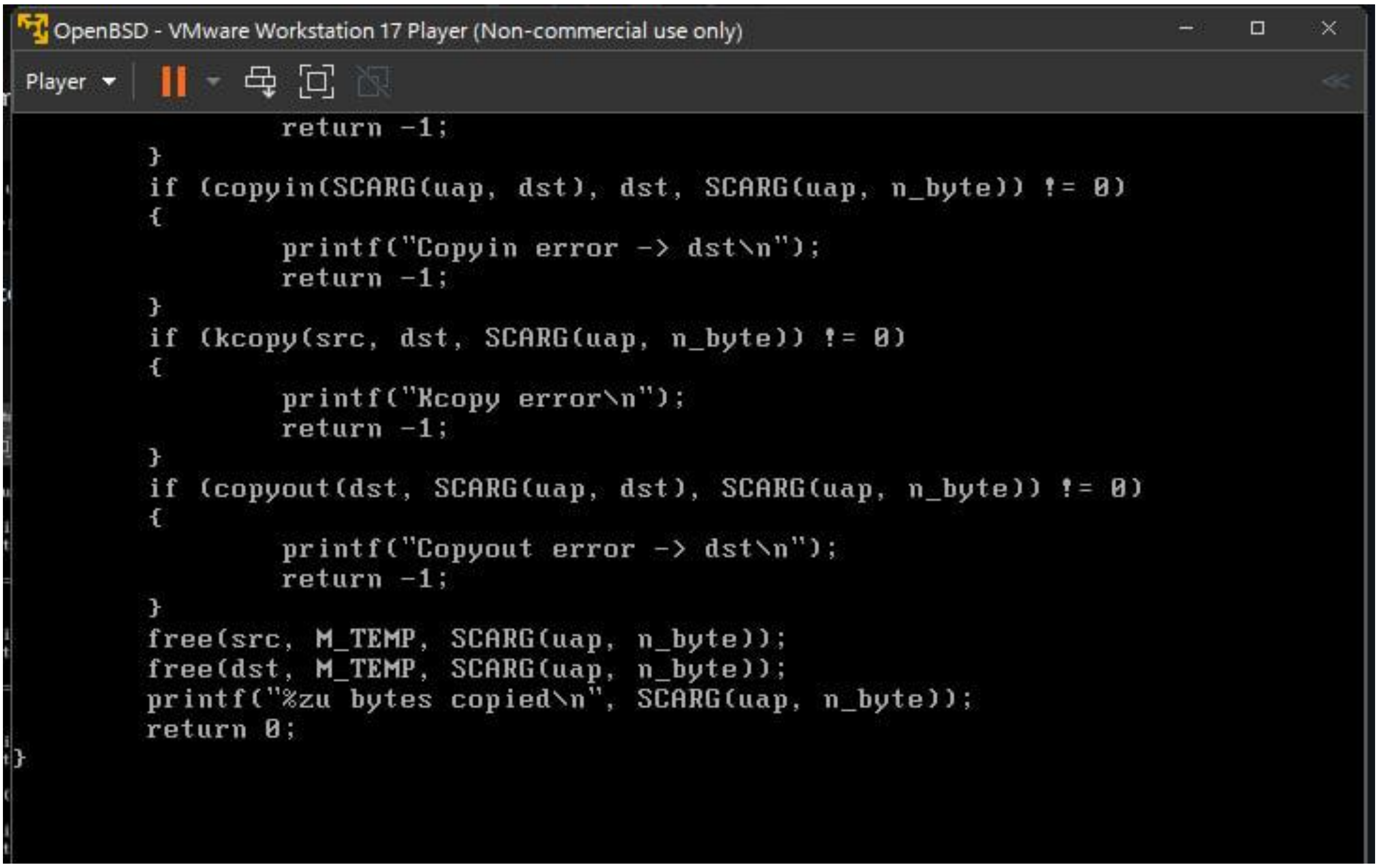

OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

```
int
sys_khello(struct proc* p, void* v, register_t* retval)
{
        struct sys_khello_args* uap = v;
        char* kmsg = malloc(120, M_TEMP, M_WAITOK);
        copyinstr(SCARG(uap, msg), kmsg, 100, NULL);
        printf("Message from kernel: %s", kmsg);
        free(kmsg, M_TEMP, 120);
        return 0;
}

int
sys_kcp(struct proc* p, void* v, register_t* retval)
{
        struct sys_kcp_args* uap = v;
        if (SCARG(uap, src) == NULL)
        {
                printf("Invalid src\n");
                return -1;
        }
        if (SCARG(uap, dst) == NULL)
        {
                printf("Invalid dst\n");
```

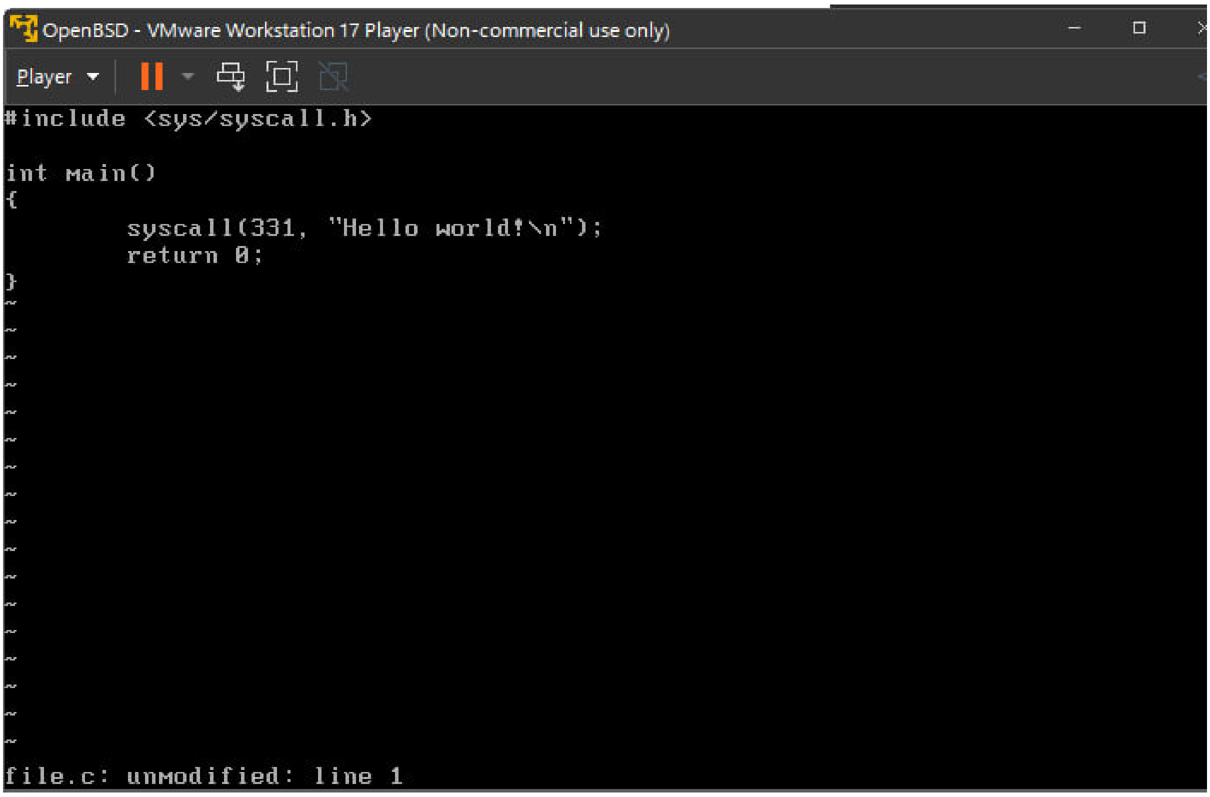

OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

```
        }
        if (SCARG(uap, dst) == NULL)
        {
                printf("Invalid dst\n");
                return -1;
        }
        void* src = malloc(SCARG(uap, n_byte), M_TEMP, M_WAITOK);
        if (src == NULL)
        {
                printf("Malloc error -> src\n");
                return -1;
        }
        void* dst = malloc(SCARG(uap, n_byte), M_TEMP, M_WAITOK);
        if (dst == NULL)
        {
                printf("Malloc error -> dst\n");
                return -1;
        }
        if (copyin(SCARG(uap, src), src, SCARG(uap, n_byte)) != 0)
        {
                printf("Copyin error -> src\n");
                return -1;
        }
        if (copyin(SCARG(uap, dst), dst, SCARG(uap, n_byte)) != 0)
```


OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

```
                return -1;
        }
        if (copyin(SCARG(uap, dst), dst, SCARG(uap, n_byte)) != 0)
        {
                printf("Copyin error -> dst\n");
                return -1;
        }
        if (kcopy(src, dst, SCARG(uap, n_byte)) != 0)
        {
                printf("Kcopy error\n");
                return -1;
        }
        if (copyout(dst, SCARG(uap, dst), SCARG(uap, n_byte)) != 0)
        {
                printf("Copyout error -> dst\n");
                return -1;
        }
        free(src, M_TEMP, SCARG(uap, n_byte));
        free(dst, M_TEMP, SCARG(uap, n_byte));
        printf("%zu bytes copied\n", SCARG(uap, n_byte));
        return 0;
}
```

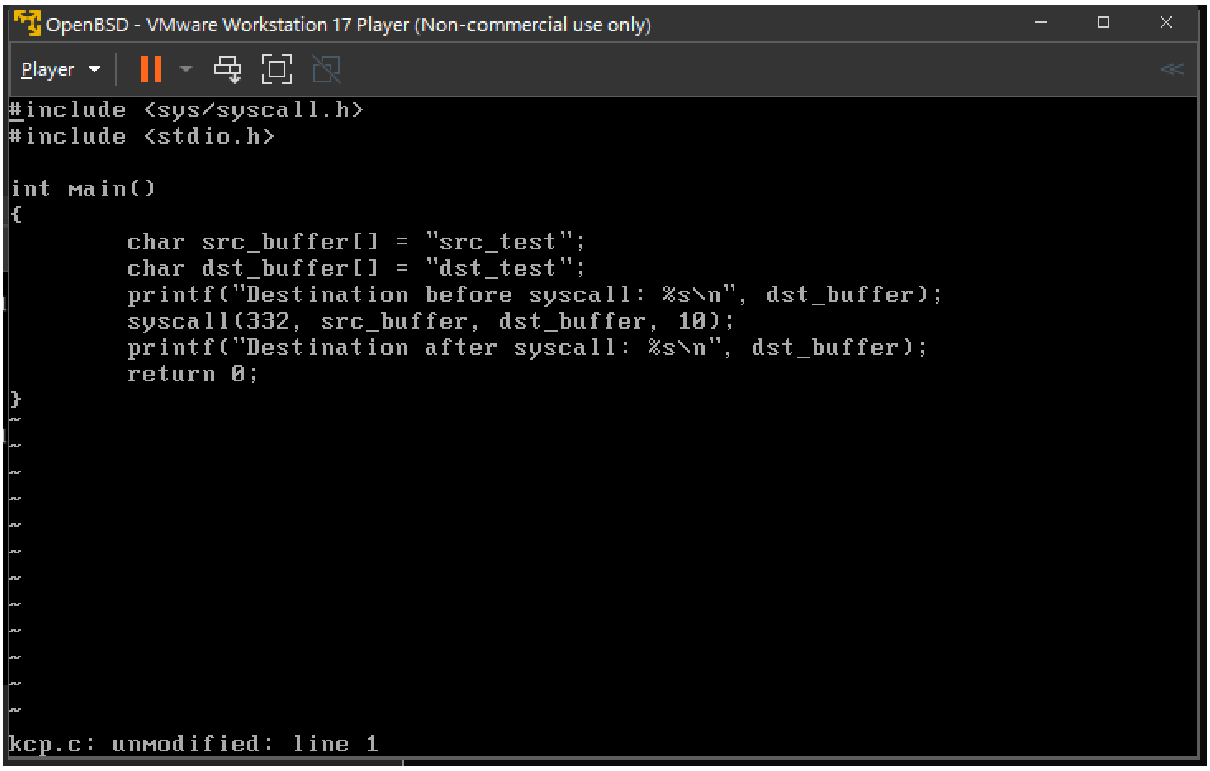
/root/file.c


OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

```
#include <sys/syscall.h>

int main()
{
        syscall(331, "Hello world!\n");
        return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
file.c: unmodified: line 1
```

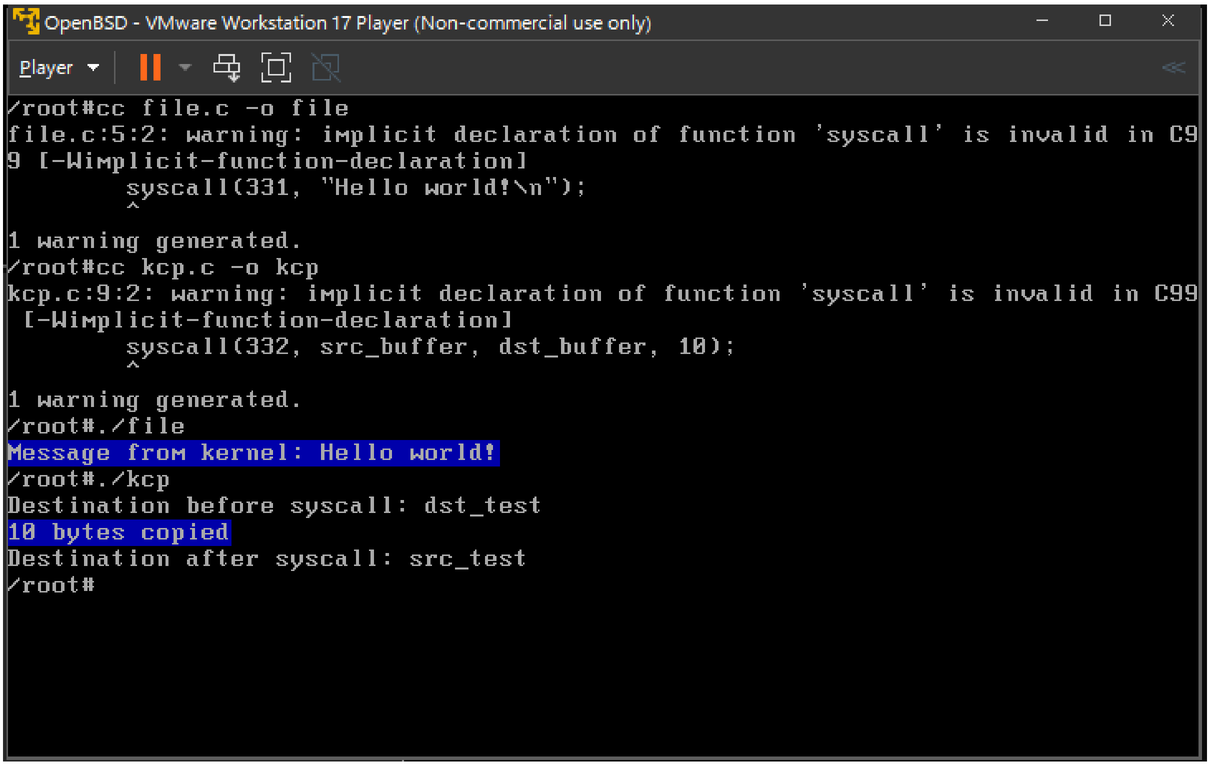
/root/kcp.c


OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

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

int main()
{
        char src_buffer[] = "src_test";
        char dst_buffer[] = "dst_test";
        printf("Destination before syscall: %s\n", dst_buffer);
        syscall(332, src_buffer, dst_buffer, 10);
        printf("Destination after syscall: %s\n", dst_buffer);
        return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
kcp.c: unmodified: line 1
```

Compilari & executari


OpenBSD - VMware Workstation 17 Player (Non-commercial use only)
Player

```
/root#cc file.c -o file
file.c:5:2: warning: implicit declaration of function 'syscall' is invalid in C9
9 [-Wimplicit-function-declaration]
        syscall(331, "Hello world!\n");
        ^

1 warning generated.
/root#cc kcp.c -o kcp
kcp.c:9:2: warning: implicit declaration of function 'syscall' is invalid in C99
 [-Wimplicit-function-declaration]
        syscall(332, src_buffer, dst_buffer, 10);
        ^

1 warning generated.
/root#./file
Message from kernel: Hello world!
/root#./kcp
Destination before syscall: dst_test
10 bytes copied
Destination after syscall: src_test
/root#
```