



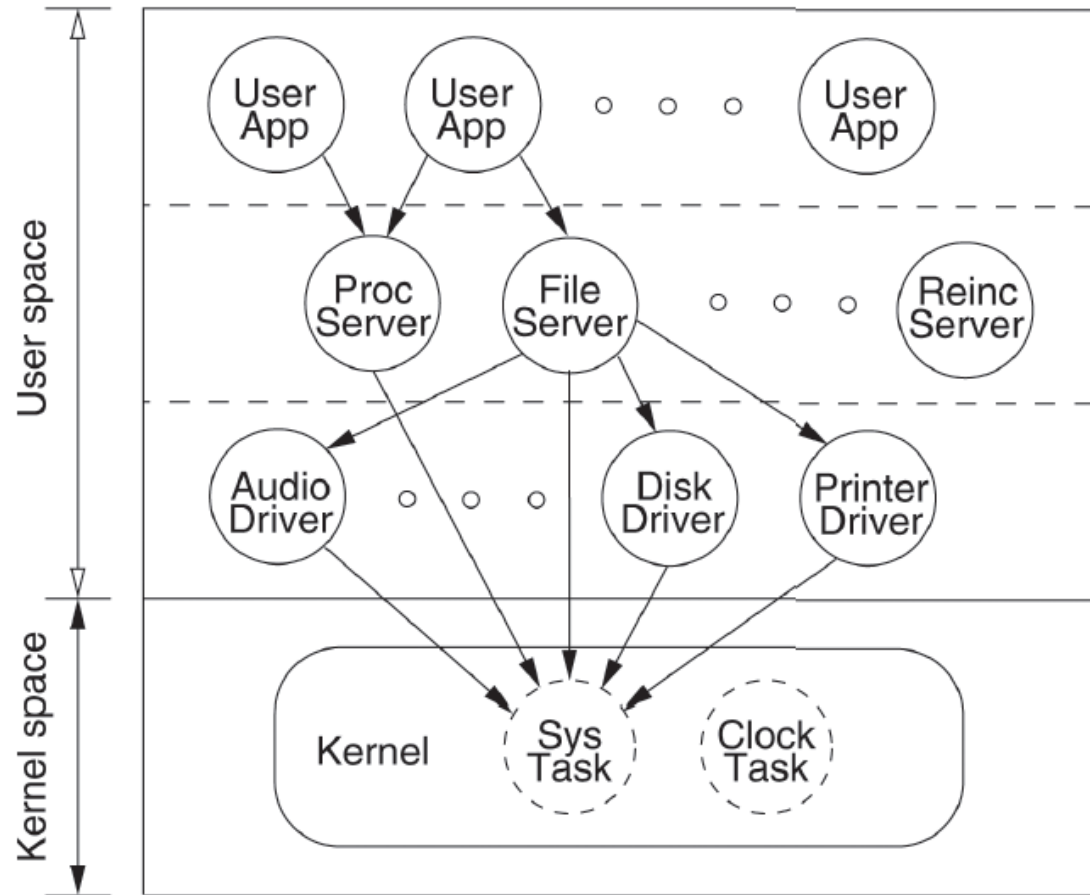
MINIX 3

MINIX 3

- Arquitectura
- Comunicación
- Llamadas al sistema
- Generar un nuevo kernel

MINIX 3 - ARQUITECTURA

- Micro-kernel
- Servidores
 - Procesos (Proc – PM)
 - Archivos (File – VFS)
 - Reencarnación



MINIX 3 - COMUNICACIÓN

- Pasaje de mensajes
- Tipos de mensajes: definidos en `/usr/src/include/minix/ipc.h`
- Ejemplos

m_source	m_source	m_source
m_type (mess_1)	m_type (mess_2)	m_type (mess_3)
m1_i1	m2_i1	m3_i1
m1_i2	m2_i2	m3_i2
m1_i3	m2_i3	m3_p1
m1_p1	m2_l1	m3_ca1
m1_p2	m2_l2	
m1_p3	m2_p1	

MINIX 3 - COMUNICACIÓN

```
typedef struct {  
    endpoint_t m_source; /* who sent the message */  
    int m_type;           /* what kind of message is it */  
    union {  
        mess_1 m_m1;  
        mess_2 m_m2;  
        mess_3 m_m3;  
        mess_4 m_m4;  
        mess_5 m_m5;  
        mess_7 m_m7;  
        mess_8 m_m8;  
        mess_6 m_m6;  
        mess_9 m_m9;  
    } m_u;  
} message;
```

MINIX 3 – LLAMADA AL SISTEMA

- Pasos para agregar una nueva llamada al sistema
 1. Incrementar la cantidad de llamadas al sistema. Identificar la nueva llamada.
 2. Elegir el servidor que va a incluir la llamada. (PM o VFS)
 3. Definir el prototipo de la llamada en el servidor correspondiente.
 4. Incluir la definición de la llamada en la tabla del servidor.
 5. Agregar la funcionalidad.
 6. Crear la librería.
 7. Recompilar el kernel.

MINIX 3 – LLAMADA AL SISTEMA

- Archivos a modificar
 - Callnr.h: definición de la cantidad de llamadas e identificación cada una de las llamadas al sistema. Ubicado en /usr/src/include/minix
 - Proto.h: definición de los prototipos del servidor correspondiente.
 - /usr/src/servers/vfs
 - /usr/src/servers/pm
 - Table.c: definición de las llamadas al sistema del servidor correspondiente.

MINIX 3 – LLAMADA AL SISTEMA: EJEMPLO

1. Editar el archivo /usr/src/include/minix/callnr.h
 1. Incrementar la cantidad de llamadas al sistema. Puede que tenga el valor **#define NCALLS 114** a **#define NCALLS 115**
 2. Definir la nueva llamada al sistema. #define NOMBRE número. Por ejemplo: #define **MENSAJE** 114
2. Elegir el servidor. Por ejemplo en VFS.
3. Definir el prototipo de la llamada en el servidor correspondiente.
 1. En el archivo /usr/src/servers/vfs/proto.h, agregar el prototipo `_PROTOTYPE(int nombre, (void))`. Por ejemplo: `int do_mensaje (void)`
4. Incluir la llamada en la tabla del servidor.
 1. En el archivo /usr/src/servers/vfs/table.c, en la línea correspondiente agregar **do_mensaje**, //línea asociada al 114.
 2. En el archivo /usr/src/servers/pm/table.c, en la línea correspondiente agregar **no_sys**, //línea asociada al 114.

MINIX 3 – LLAMADA AL SISTEMA: EJEMPLO

5. Código de la funcionalidad del nuevo system call **MENSAJE**, **do_mensaje**

```
• int do_mensaje()  
{  
    printf("Este mensaje se muestra  
    desde el system call MENSAJE \n");  
  
    return(0);  
}
```

- Ejemplo de parámetros de entrada y salida en el caso de utilizar el servidor VFS:

```
A = m_in.m1_i2  
B = m_in.m1_i3  
m_out.m1_i1 = c
```

MINIX 3 – LLAMADA AL SISTEMA: EJEMPLO

Programa fuente con invocación al nuevo system call. Ejemplo.c

```
#include<mosmensaje.h>
```

```
Main(){  
    mosmensaje();  
    ...  
    return 0;  
}
```

Librería, incluida en
[**/usr/include/mosmensaje.h**](#)

```
int mosmensaje(void)  
{  
    message m;  
    _syscall(VFS_PROC_NR,  
            MENSAJE, &m)  
    return 0;  
}
```

servidor

Nombre del
System call.

MINIX 3 – LLAMADA AL SISTEMA: EJEMPLO

6. Crear las librerías

- `cd /usr/src/tools`
- `make includes`

7. Recompilar el kernel

- `cd /usr/src/tools`
- `make hdboot`