



**UASLP**

Universidad Autónoma  
de San Luis Potosí



**FACULTAD DE  
INGENIERÍA**

## **Práctica 4. Banderas de visualización.**

Sistemas operativos B.

Profesora: M.I. Ortiz Hernández Marcela.

grupo: 2402-03.

Semestre: 2019-2020/II.

Cantu Olivares Pedro de Jesus..

**21/Mayo/2020.**

## 1.- Resumen:

Realizar la modificación de la práctica 3, Paginación por demanda pura con algoritmo de reemplazo de tal forma que cuando se ejecute un programa se pueda elegir la salida a visualizar en la terminal. Puntos a implementar

## 2.- Instrucciones:

1. Modificar la orden de ejecución(línea de comandos en terminal), agregando un argumento que especificará la salida a visualizar utilizar cuando se ejecute un programa, se muestra a continuación: La orden de ejecución de Nachos en su forma original es de la siguiente manera: `$/nachos -x ../test/nombre_programa` Se deberá realizar la siguiente modificación `$/nachos -[C ó M ó F -o R] -x ../test/nombre_programa` donde `nombre_programa` es `halt`, `matmult` o `sort`. Las opciones deben especificarse en ese orden. De esta manera, el usuario es capaz de seleccionar la salida que deberá mostrarse en la terminal.

2. Descripción de las nuevas opciones El argumento adicional puede ser la letra C, ó M, ó F, ó R mayúscula, dependiendo del argumento que se especifique será la salida que se va a visualizar de la práctica. a) La opción -C indica que solo se va a visualizar como salida el tamaño del proceso y la cantidad de páginas que lo forman, además, del vpn de cada dirección lógica separadas por comas (es decir, se visualiza la cadena de referencia). b) La opción -M indica que se va a visualizar el mapeo de cada dirección lógica a física que generó fallo de página (dirección lógica, vpn, offset, dirección física) y el número de fallo. c) La opción -F indica que se va a visualizar como salida el tamaño del proceso, la cantidad de páginas que lo forman, además, del vpn de las direcciones lógicas que generaron fallo de página. d) La opción -R indica que solo mostrará el resumen de la máquina.

3. Es necesario validar los argumentos de entrada y marcar los errores en caso de que se presenten. Es decir, errores de sintaxis.

**3.- Liste todos los archivos que se modificaron para esta práctica y una breve descripción de modificación:**

**1.- archivo modificado: ./nachos/userprog/main.cc:**

**Descripción:**

Declaración de variable para la bandera que se usará para imprimir el flujo del programa.

**Código:**

```
/******  
Practica 4  
*****/  
char *comando_Fifo;//guarda el tipo de impresion segun el  
comando
```

**2.-archivo modificado: ./nachos/threads/main.cc:**

**Descripción:**

Se agrega una variable(comando\_Fifo) externa que representa a la cadena para la bandera.

**Código:**

```
/******  
Prsctica 4 variable externa para los comandos agregados  
en la practica.  
*****/  
extern char* comando_Fifo;
```

**3.-archivo modificado: ./nachos/threads/main.cc:**

**Descripción:**

Entradas para las banderas(C,F,M,R) junto con el comando -x, para ejecutar.

**Código:**

```
#ifdef USER_PROGRAM  
    if(!strcmp(*argv, "-C"))  
    {  
        ASSERT(argc > 2);  
        if(!strcmp(*(argv+1), "-x"))  
        {  
            comando_Fifo = "-C";  
            StartProcess(*(argv + 2));  
            argCount = 3;  
        }  
    }
```

```

        else
        {
            printf("\nSintaxis incorrecta, intenta
con:\n\t./nachos -C -x nombre_Programa.\n");
            interrupt->Halt();
        }

    }else if(!strcmp(*argv, "-M"))
    {
        ASSERT(argc > 2);
        if(!strcmp(*(argv+1), "-x"))
        {
            comando_Fifo = "-M";
            StartProcess(*(argv + 2));
            argCount = 3;
        }
        else
        {
            printf("\nSintaxis incorrecta, intenta
con:\n\t./nachos -M -x nombre_Programa.\n");
            interrupt->Halt();
        }
    }
    else if(!strcmp(*argv, "-F"))
    {
        ASSERT(argc > 2);
        if(!strcmp(*(argv+1), "-x"))
        {
            comando_Fifo = "-F";
            StartProcess(*(argv + 2));
            argCount = 3;
        }
        else
        {
            printf("\nSintaxis incorrecta, intenta
con:\n\t./nachos -F -x nombre_Programa.\n");
            interrupt->Halt();
        }
    }
    else if(!strcmp(*argv, "-R"))

```

```

{
    ASSERT(argc > 2);
    if(!strcmp(*(argv+1), "-x"))
    {
        comando_Fifo = "-R";
        StartProcess(*(argv + 2));
        argCount = 3;
    }
    else
    {
        printf("\nSintaxis incorrecta, intenta
con:\n\t./nachos -R -x nombre_Programa.\n");
        interrupt->Halt();
    }
}
else if (!strcmp(*argv, "-x")) {           // run
a user program
    ASSERT(argc > 1);
    comando_Fifo = "-x";
    StartProcess(*(argv + 1));
    argCount = 2;
} else if (!strcmp(*argv, "-c")) {         // test
the console
    if (argc == 1)
        ConsoleTest(NULL, NULL);
    else {
        ASSERT(argc > 2);
        ConsoleTest(*(argv + 1), *(argv + 2));
        argCount = 3;
    }
    interrupt->Halt();           // once we start the
console, then

                                // Nachos will loop forever
waiting

                                // for console input
}
#endif // USER_PROGRAM

```

#### 4.-archivo modificado: ./nachos/machine/translate.cc:

##### Descripción:

Definición de la impresión según la bandera en translate.cc en la clase Machine, en su método Translate(Machine::Translate()).

### Código:

```
/******  
Practica 4  
*****/  
if(!strcmp(comando_Fifo, "-M"))  
{  
    printf("\nFallo #  
%d\n", stats->numPageFaults +1);  
    printf("Direccion Virtual(Logica):  
%d\n", virtAddr);  
    printf("offset(Desplazamiento)  
calculado: %d\n", offset);  
    printf("vpn calculado: %d\n", vpn);  
}  
if(!strcmp(comando_Fifo, "-F"))  
{  
    printf("vpn -> Fallo# %d:  
%d\n", stats->numPageFaults +1, vpn );  
}
```

### 5.-archivo modificado: ./nachos/userprog/addrSpace.cc:

#### Descripción:

Definicion de la impresion segun la bandera en addrSpace.cc en la clase AddrSpace y su constructor.

### Código:

```
/******  
Practica 0 :  
*****/  
if(!strcmp(comando_Fifo, "-C") ||  
!strcmp(comando_Fifo, "-F") )//Practica 4  
{  
    printf("\nTamaño del proceso: %d  
Bytes.\n", size); //imprime el tamaño del proceso.  
    printf("\nNumero de paginas para el proceso:  
%d.", numPages); //imprime el numero de paginas necesarias  
para cargar el proceso.  
}
```

#### **4. Conclusiones**

Las variables externas funcionan para usarlas en varios archivos incluso si estos están en otro directorio. Es una forma inteligente el ejecutar los programas con banderas para especificar el modo que funcionara el programa y además tener mayor control sobre el mismo.