

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Ingeniería en Computación

Sistemas Operativos

Proyecto 4: (Micro) sistema de archivos

Profesor: Gunnar Eyal Wolf Iszaevich

Alumnos:

Bárcenas Avelar, Jorge Octavio

Reza Chavarria, Sergio Gabriel

Entrega: 26 de mayo del 2020

Proyecto 4: (Micro) sistema de archivos

Objetivo

A partir de los conocimientos obtenidos, poder implementar un sistema de archivos, a partir de una especificación y caso de referencia.

Desarrollo

Herramientas utilizadas

En lenguaje de desarrollo utilizado fue Python (Versión 3.8.2). Se trabajaron en 2 sistemas operativos.

- Linux Mint 19.1
- Ubuntu 18.04.4

Las bibliotecas utilizadas en este proyecto fueron las siguientes

- Mmap: Utilizado para el manejo de mapas de memoria en lugar de dar lectura a contenidos directamente. Este fue utilizado para manejar el micro sistema de archivos “FiUnamFS”
- Os: Biblioteca utilizada para escribir o leer cadenas de caracteres de los archivos.

Procesos

- Clase BloquePrincipal: Esta clase es utilizada para obtener la información del primer cluster. Se utilizó mmap para obtener acceso a esta información, usando decodificación en utf-8.
- Clase Entrada: Esta clase obtiene las entradas del sistema y obtiene igual la información a través de la decodificación.
- Clase SisArch: Clase del programa la cual será encargada del manejo de las funciones necesarias para realizar el listado de archivos, copia de archivos, eliminación y desfragmentación.
 - Inodo: Utilizado para obtener todos los inodos que el mini sistema contiene. Tiene uso del bloque principal, utilizado para calcular el tamaño usado, y así obtiene la entrada, si el nombre de la entrada es diferente del nombre de

entrada 'Xx.xXx.xXx.xXx.', se guardará en una lista de Inodos. Esta lista se utilizará en otras funciones.

- Buscar: Se realizará una búsqueda lineal a partir de las entradas encontradas, ayudándonos del tamaño de estos. Si el nombre del archivo buscado coincide con el archivo de la iteración, este archivo será devuelto.
- Registrar: Esta función hace un recorrido de la información del micro sistema, si encuentra un espacio que se encuentre como entrada no utilizada, se guardarán los datos del nuevo archivo en formato adecuado. El nombre alineado a la derecha, y lo rellenará con espacios para el relleno completo, se obtendrá el tamaño del archivo, que igual se guardará con relleno de 0 a la izquierda, se añade el cluster, la fecha de creación y modificación, en el formato declarado, todos estos datos se guardarán en el micro sistema en bytes.
- Ls: Imprime la información de todos los inodos encontrados, con salida del nombre, cluster, tamaño, mes, día, año, hora de la modificación.
- Rm: Se realizará una búsqueda con el nombre del archivo a remover. Si no lo encuentra se le avisará al usuario, si es que lo encuentra se obtendrá el tamaño de la entrada y el número de dirección. Y a partir de esta información se pondrá el nombre de la entrada no usada.
- Cpout: Función para que un archivo sea copiado al sistema. Realiza una búsqueda para revisar si el archivo existe, si no existe se le notificará al usuario. Si es que encuentra el archivo obtiene la información de este, abre el archivo y esté lo escribe en el lugar en donde se encuentre ejecutándolo.
- Cpin: Función que guardará archivos en el micro sistema. Primero revisará si existe el archivo, si este existe se revisará el tamaño de este, y si es menor a 15 se revisará si este nombre de archivo existe en el microsistema, si no es así se llamará a la función de copiado.
- Copiar: Se obtendrá la información de los inodos y serán ordenados dependiendo del tamaño del cluster. Existen 3 casos, en el que él no haya inodos, se revisará si este no es mayor al total de clústers disponibles. Si hay datos y puede que se guarden entre archivos se podrá añadir entre esos

archivos, si no hay espacio suficiente y hay espacio al final se podrá registrar. En el caso en el que no exista suficiente espacio contiguo se le notificará al usuario

- Defrag: Función de desfragmentación del microsistema.
- Printall: Es una función que se muestra los 64 espacios usados para entradas.
- FormatoFecha: Uso de un diccionario para el formato de los meses y el desglosamiento de la información de la fecha.

Pruebas

python3 proy_4.py -help

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py -help
ls -> Mostrar directorio
printall -> Mostrar los 64 espacios para entradas
rm [FILE] -> Eliminar Archivo
cpout [FILE] "DIR" -> Copiar archivo a nuestro sistema
cpin [FILE] -> Copiar archivo al sistema
defrag -> Desfragmentación
```

python3 proy_4.py ls

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py ls
Nombre Cluster Tamaño Fecha y hora de Creacion Fecha y hora de Modificación
README.org 5 29528 May 04 2020 13:45:00 May 04 2020 13:45:00
logo.png 37 174875 May 04 2020 13:45:00 May 04 2020 13:45:00
datetime.txt 352 63 May 04 2020 13:45:00 May 04 2020 13:45:00
mensaje.png 353 53509 May 04 2020 13:45:00 May 04 2020 13:45:00
```

python3 proy_4.py cpin <File>

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ ls
fif.py fiunamfs.img fsm.py proy_4.py proyecto3.py pruebas_proy_4.py
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py cpin proyecto3.py
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py ls
Nombre Cluster Tamaño Fecha y hora de Creacion Fecha y hora de Modificación
README.org 5 29528 May 04 2020 13:45:00 May 04 2020 13:45:00
proyecto3.py 10 9033 May 21 2020 18:21:00 May 07 2020 02:27:51
logo.png 37 174875 May 04 2020 13:45:00 May 04 2020 13:45:00
datetime.txt 352 63 May 04 2020 13:45:00 May 04 2020 13:45:00
mensaje.png 353 53509 May 04 2020 13:45:00 May 04 2020 13:45:00
```

python3 proy_4.py rm <File>

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py ls
Nombre Cluster Tamaño Fecha y hora de Creacion Fecha y hora de Modificación
README.org 5 29528 May 04 2020 13:45:00 May 04 2020 13:45:00
proyecto3.py 10 9033 May 21 2020 18:21:00 May 07 2020 02:27:51
logo.png 37 174875 May 04 2020 13:45:00 May 04 2020 13:45:00
datetime.txt 352 63 May 04 2020 13:45:00 May 04 2020 13:45:00
mensaje.png 353 53509 May 04 2020 13:45:00 May 04 2020 13:45:00
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py rm proyecto3.py
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py ls
Nombre Cluster Tamaño Fecha y hora de Creacion Fecha y hora de Modificación
README.org 5 29528 May 04 2020 13:45:00 May 04 2020 13:45:00
logo.png 37 174875 May 04 2020 13:45:00 May 04 2020 13:45:00
datetime.txt 352 63 May 04 2020 13:45:00 May 04 2020 13:45:00
mensaje.png 353 53509 May 04 2020 13:45:00 May 04 2020 13:45:00
```

python3 proy_4.py cpout <File> <Dir>

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ ls
fiunamfs.img proy_4.py proyecto3.py pruebas_proy_4.py
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py cpout logo.png "/home/jorge/Desktop/Proyectos_Sistop/proyecto4"
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ ls
fiunamfs.img logo.png proy_4.py proyecto3.py pruebas_proy_4.py
```

python3 proy_4.py defrag

python3 proy_4.py printall

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py printall
entrada + archivo + cluster + fecha_creación + fecha_modificación
0 + README.org + 5 + May 04 2020 13:45:00 + May 04 2020 13:45:00
1 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
2 + logo.png + 37 + May 04 2020 13:45:00 + May 04 2020 13:45:00
3 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
4 + datetime.txt + 352 + May 04 2020 13:45:00 + May 04 2020 13:45:00
5 + mensaje.png + 353 + May 04 2020 13:45:00 + May 04 2020 13:45:00
6 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
7 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
8 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
9 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
10 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
```

```
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py defrag
Defrag Complete
jorge@pc-joba:~/Desktop/Proyectos_Sistop/proyecto4$ python3 proy_4.py printall
entrada + archivo + cluster + fecha_creación + fecha_modificación
0 + README.org + 5 + May 04 2020 13:45:00 + May 04 2020 13:45:00
1 + logo.png + 37 + May 04 2020 13:45:00 + May 04 2020 13:45:00
2 + datetime.txt + 352 + May 04 2020 13:45:00 + May 04 2020 13:45:00
3 + mensaje.png + 353 + May 04 2020 13:45:00 + May 04 2020 13:45:00
4 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
5 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
6 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
7 + Xx.xXx.xXx.xXx. + 0 + 0000000000000000 + 0000000000000000
```

Conclusiones

Bibliografía

- Rico, E. (2018). *Mmap — Archivos de mapas de memoria*. Encontrado el 19 de mayo, 2020, en <https://rico-schmidt.name/pymotw-3/mmap/>
- Programación en Python (Desconocido) 7.2. *Manipulación de archivos*. Encontrado el 19 de mayo, 2020, en <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion7/archivos.html>