Universidad Autónoma de Chihuahua

Sistemas Operativos I

Proyecto a Mitad de Semestre

Jadir Alexis Levario Ojeda – 362243

Alejandro Morales Rodríguez – 357796

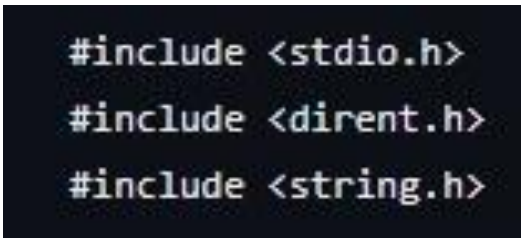
Ariana Janeth Franco Ríos – 357756

Andree Javier Ordaz Chavira - 360844

Chihuahua, Chih 15/11/2023

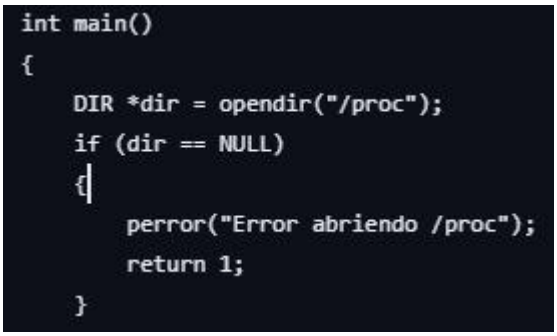
¿Cómo funciona nuestro código?

Pues….



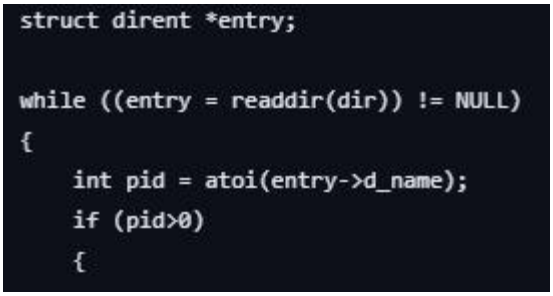
```
#include <stdio.h>
#include <dirent.h>
#include <string.h>
```

Inclusión de las bibliotecas: Estas son bibliotecas estándar de C que son necesarias para realizar operaciones de entrada/salida (stdio.h), para manipular directorios (dirent.h) y también para realizar cadenas (string.h).



```
int main()
{
    DIR *dir = opendir("/proc");
    if (dir == NULL)
    {
        perror("Error abriendo /proc");
        return 1;
    }
```

Apertura del directorio "/proc": Se abre el directorio /proc utilizando la función "opendir". Si se encuentra algún error durante la apertura del directorio, se mostrará un mensaje de error.



```
struct dirent *entry;

while ((entry = readdir(dir)) != NULL)
{
    int pid = atoi(entry->d_name);
    if (pid>0)
    {
```

Iteración sobre elementos del directorio /proc: Se utiliza un bucle while para iterar sobre los elementos del directorio /proc utilizando la función "readdir". Se convierte el nombre del directorio en un numero entero PID usando un atoi.

```
char status_path[256];
snprintf(status_path, sizeof(status_path), "/proc/%d/status", pid);

FILE *status_file = fopen(status_path, "r");
if (status_file != NULL)
```

Construcción de la ruta y apertura del archivo status: En ese punto se construye la ruta al archivo /proc/PID/status utilizando snprintf. Luego de esto, se abre el archivo como modo lectura “r” utilizando fopen, si la apertura fue exitosa, se procederá a leer el archivo.

```
char line[256];
char name[256];

while (fgets(line, sizeof(line), status_file) != NULL)
{
    if (sscanf(line, "Name:\t%s", name) == 1)
    {
        break;
    }
}
```

Lectura de líneas del archivo status y extracción del nombre: Se utiliza un bucle while y un fgets para leer las líneas del archivo status. Luego, se utiliza un scanf para extraer el nombre del proceso de la línea correspondiente al campo “Name”.

```
                int is_kernel = strstr(line, "VmKernel") != NULL;

                printf("ID del proceso: %d\n", pid);
                printf("Nombre del proceso: %s\n", name);
                printf("Tipo de proceso: %s\n", is_kernel ? "Kernel" : "Usuario");
                printf("============================\n");

                fclose(status_file);
            }
        }
    }

    closedir(dir);

    return 0;
}
```

Determinación si es un proceso de kernel y muestra de información: Se utiliza un strstr para determinar si la línea contiene la cadena “VmKernel”, lo que indica si el proceso es de kernel. Luego se mostrará la información del proceso, lo cual incluye el ID del proceso, el nombre y el tipo (kernel o usuario), y si cierra el archivo status utilizando fclose.

Una screenshot de nuestro programa funcionando

```
Tipo de proceso: Usuario
============================
ID del proceso: 26451
Nombre del proceso: kworker/0:2-events
Tipo de proceso: Usuario
============================
ID del proceso: 26452
Nombre del proceso: kworker/u6:4-events_unbound
Tipo de proceso: Usuario
============================
ID del proceso: 26455
Nombre del proceso: systemd-userwor
Tipo de proceso: Usuario
============================
ID del proceso: 26520
Nombre del proceso: jad2
Tipo de proceso: Usuario
============================
[root@localhost-live liveuser]# S
```

¿Cómo saber cuánta memoria usamos?

```c
FILE *meminfo_file = fopen("/proc/meminfo", "r");
if (meminfo_file != NULL)
{
    char line[256];

    while (fgets(line, sizeof(line), meminfo_file) != NULL)
    {
        printf("%s", line);
    }
    fclose(meminfo_file);
}
```

Mostrar información de la memoria del sistema: Utilizamos un fopen para abrir el archivo “meminfo”, si se logra abrir el archivo, este procederá a leerlo; después de eso utilizará un fgets para leer cada línea del archivo “meminfo”, procedente a esto se imprime cada línea que contiene información sobre la memoria del sistema.