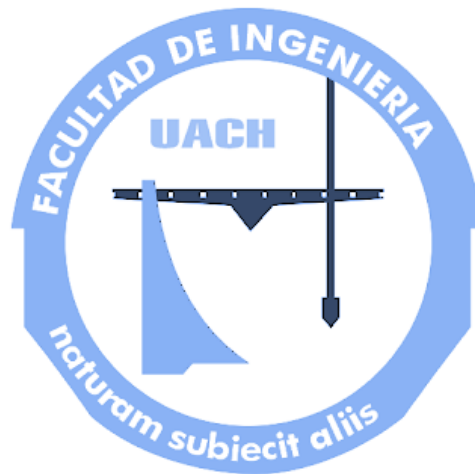


“Universidad Autónoma De Chihuahua”



“LECTURA DE PROCESOS DEL KERNEL”

Materia: Sistemas Operativos I

Profesor: Ivan Miguel Chavero Jurado

Fecha de entrega: 23/10/2024

Integrantes Del Equipo:

- **Ashly Alessandra Guzmán Ayala 367756**
- **Jaime Rey Mendoza 348691**
- **Tadeo Andrei Nieto Paz 361904**



CODIGO EN C PARA LECTURA DE PROCESOS DEL KERNEL

El código tiene como objetivo listar los procesos del sistema, obteniendo información de los archivos `/proc/[PID]/cmdline` y `/proc/[PID]/stat`. Además, identifica si los procesos pertenecen al kernel o al usuario.

El programa abre el directorio `/proc`, que es un directorio virtual en Linux que contiene información sobre todos los procesos del sistema.

Para cada proceso encontrado, que se identifica por un subdirectorio numérico (el ID de proceso o PID), abre dos archivos específicos:

- **cmdline**: que contiene el comando que inició el proceso.
- **stat**: que contiene información sobre el estado del proceso.

Explicación:

```
int main (int argc, char *argv[]) {
    struct dirent *entry;
    DIR *pDir;
    char proc_dir[] = "/proc";
    char proc_id_dir[255];
    char proc_id_dir1[255];
    char cmdline[500];
    char statinfo[500];
```

proc_dir[]: Define el directorio `/proc`, que contiene subdirectorios para cada proceso identificado por su PID.

cmdline y **statinfo**: Almacenan el contenido de los archivos correspondientes a cada proceso.

```
pDir = opendir (proc_dir);
if (pDir == NULL) {
    printf ("Cannot open directory UFF '%s'\n", argv[1]);
    return 1;
}
```

pDir = opendir (proc_dir): Verifica si existe el `/proc` y pone `pDir` al inicio del Directorio de procesos.

```
while ((entry = readdir(pDir)) != NULL) {
    if ( ! isdigit(entry->d_name[0]) )
        continue;
```

Mediante un while se recorren todos los procesos del directorio y revisa si es nulo o no.

Si no es nulo verifica si el nombre del proceso es un número.

```
    sprintf(proc_id_dir, "/proc/%s/cmdline", entry->d_name);
    get_process_cmd(proc_id_dir, cmdline);
    sprintf(proc_id_dir1, "/proc/%s/stat", entry->d_name);
    get_process_stat(proc_id_dir1, statinfo);
```

Cuando sí sea un proceso almacenamos la dirección del proceso, se va a guardar "/proc/entry->d_name/cmdline" dentro de **proc_id_dir**, lo mismo con el stat.

get_process_cmd/stat(proc_id_dir, cmdline/stat): Accedemos al cmdline/stat a través de una función que manda como parámetro la dirección del proceso (proc_id_dir).

Dentro de la función:

```
void get_process_stat(char *filename, char *statinfo) {
    FILE *fp;
    char *line1 = NULL;
    size_t len1 = 0;
    statinfo[0] = '\0';

    // Open file in read-only mode
    fp = fopen(filename, "r");

    // If file opened successfully, then print the contents
    if ( fp == NULL ) {
        printf("Error abriendo archivo %s\n", filename);
        exit(2);
    }

    while(getline(&line1, &len1, fp) != -1) {
        sprintf(statinfo, "%s%s", statinfo, line1);
    }

    fclose(fp);
    free(line1);
}
```

Primero se inicialización de los parámetros del archivo, se abre el archivo y con la función **getline** (que está definida dentro de stdio.h) concatenamos el archivo de stat para imprimirlo o en su caso el cmdline.

```
tam = ftell(fp);
```

esta línea dentro de la función **get_process_cmd** obtiene la posición actual del puntero del archivo y si el cmd está vacío tam va a ser igual a 0

```

printf ("\n[%s]:CMDLINE:%s STAT: %s", entry->d_name, cmdline, statinfo);
if(tam==0){
    printf("Es proceso de Kernel\n");
    printf("----->
}
else{
    printf("Es proceso de Usuario\n");
    printf("----->
}
}

```

Imprimimos la ID del proceso seguido del CMDLINE, el STAT y si el tam del cmdline es igual a 0 va a imprimir que el proceso es exclusivo del kernel.

Así se ve la salida del programa:

```

-----
[15214]:CMDLINE:/usr/lib64/firefox/firefox STAT: 15214 (Web Content) S 3231 2207 2207 0 -1 419
4560 4961 0 0 0 10 14 0 0 20 0 20 0 963140 2716893184 21014 18446744073709551615 9419346776064
0 94193468340749 140733894184704 0 0 0 0 16846850 1082134264 0 0 0 17 5 0 0 0 0 0 941934684631
28 94193468469368 94194316980224 140733894189833 140733894190142 140733894190142 1407338941931
17 0
Es proceso de Usuario
-----

[15216]:CMDLINE:/usr/lib64/firefox/firefox STAT: 15216 (Web Content) S 3231 2207 2207 0 -1 419
4560 4888 0 0 0 11 13 0 0 20 0 20 0 963143 2716893184 20628 18446744073709551615 9398188053708
8 93981881117197 140735659471312 0 0 0 0 16846850 1082134264 0 0 0 17 4 0 0 0 0 0 939818812395
76 93981881245816 93982394195968 140735659479817 140735659480126 140735659480126 1407356594831
01 0
Es proceso de Usuario
-----

[15287]:CMDLINE: STAT: 15287 (kworker/3:0-events) I 2 0 0 0 -1 69238880 0 0 0 0 0 0 0 20 0 1
0 963779 0 0 18446744073709551615 0 0 0 0 0 0 2147483647 0 0 0 0 17 3 0 0 0 0 0 0 0 0 0
0 0
Es proceso de Kernel
-----

[15301]:CMDLINE: STAT: 15301 (kworker/u48:1-gfx) I 2 0 0 0 -1 69238880 0 0 0 0 0 0 6 0 0 20 0 1
0 964771 0 0 18446744073709551615 0 0 0 0 0 0 2147483647 0 0 0 0 17 2 0 0 0 0 0 0 0 0 0
0 0
Es proceso de Kernel
-----

```

En resumen:

- El programa lista todos los procesos leyendo el directorio **/proc**.
- Para cada proceso, obtiene el **cmdline** y su estado **stat**.
- Clasifica los procesos como de usuario o de kernel.

NOTAS

Directorio: carpeta o archivo que contiene una lista de otros archivos. Un directorio es un contenedor virtual que almacena archivos y subdirectorios, organizados por su contenido, propósito o cualquier criterio que el usuario decida.

Directorio /proc: es un archivo que contiene información sobre el sistema, los procesos y los parámetros de configuración.

DATOS QUE LEE EL CÓDIGO:

Proceso de Usuario - PID: 3270 | CMDLINE: ./Newman2

PID: 3270 | STAT: 3270 (Newma2) R 2978 3270 2978 34816 3270 4194304 154 0

78 0 0 1 0 0 20 0 1 0 129855 6205440 864 18446744073709551615 4198400

4199677 140730948513296 0 0 0 0 0 0 0 0 17 0 0 0 0 0 4210120 4210788

13119488 140730948518566 140730948518575 140730948518575

140730948521967 0

SIGNIFICADO DEL STAT:

STAT: (ID DEL PROCESO)

Newman2: Nombre del proceso.

R: Estado Del Proceso(RST).

2978 (Primero Dígito): ID Del Proceso Padre.

3270 (): Id Del Grupo de procesos.

154 (): Tiempo Usuario.

Para compilar y ejecutar un código:

tadeonietopaz@vbox:~\$ cd Documentos

tadeonietopaz@vbox:~/Documentos\$ ls

a.out Newman Newman.cpp Newman.cpp.savecd

tadeonietopaz@vbox:~/Documentos\$ nano Newman2.cpp

tadeonietopaz@vbox:~/Documentos\$ g++ -o Newman2 Newman2.cpp

tadeonietopaz@vbox:~/Documentos\$./Newman2

Explorando el directorio /proc para listar procesos y sus cmdline y stat...
