

SISTEMAS OPERATIVOS 1

Brandon Andy Jefferson Tejaxún Pichiyá

202112030



Herramientas:



Repositorio: https://github.com/brandonT2002/SO1_2S2024_202112030

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/sysinfo.h>
#include <dirent.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>

// Función para obtener la información de la memoria
void obtener_memoria() {
    struct sysinfo si;
    if (sysinfo(&si) == 0) {
        printf("Total RAM: %lu MB\n", si.totalram / (1024 * 1024));
        printf("Free RAM: %lu MB\n", si.freeram / (1024 * 1024));
    } else {
        perror("sysinfo");
    }
}

// Función para verificar si un directorio es un proceso
int es_proceso(const struct dirent *entry) {
    if (entry->d_type == DT_DIR) {
        for (int i = 0; entry->d_name[i] != '\0'; i++) {
            if (!isdigit(entry->d_name[i])) {
                return 0;
            }
        }
        return 1;
    }
    return 0;
}

// Función para listar los procesos padre e hijos
void listar_procesos() {
    DIR *dir;
    struct dirent *entry;
```

```
char path[1024], line[256];
FILE *fp;

if ((dir = opendir("/proc")) == NULL) {
    perror("opendir");
    return;
}

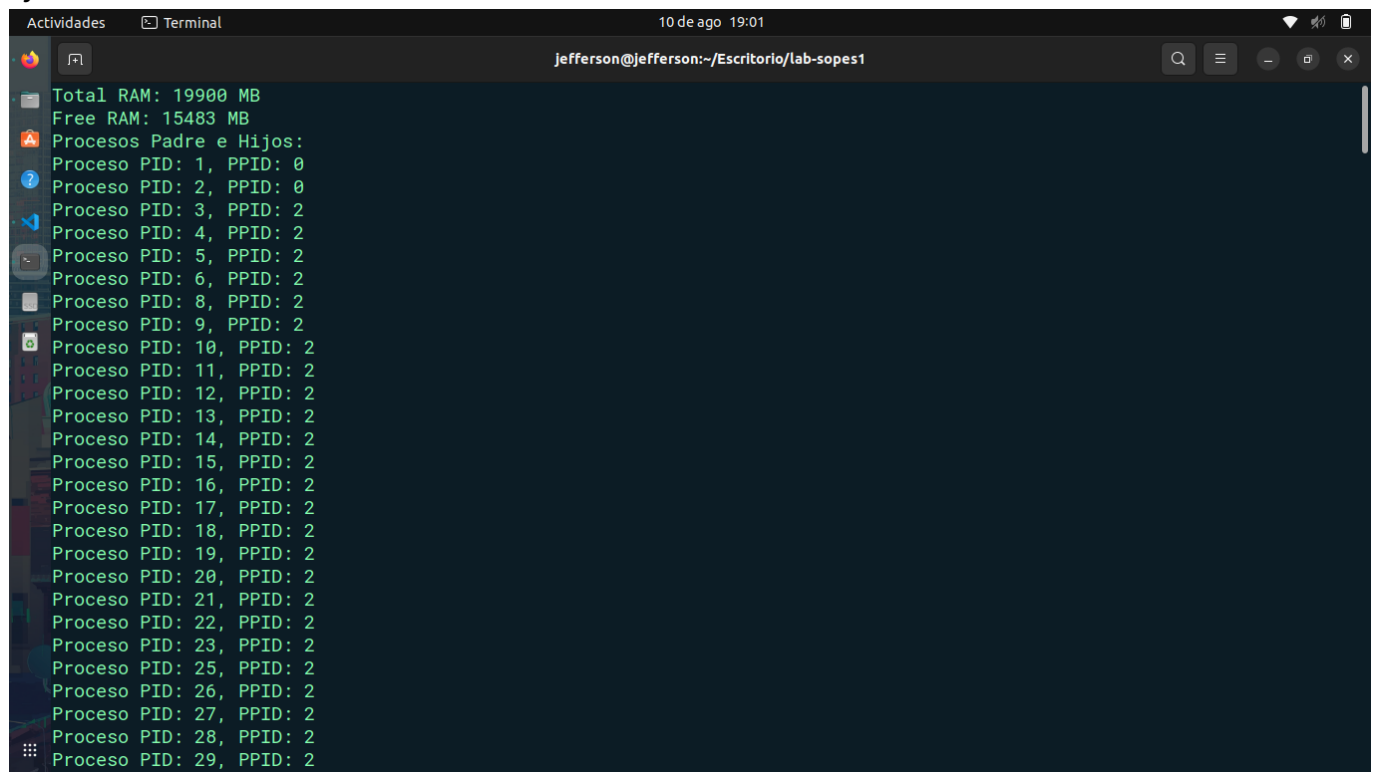
printf("Procesos Padre e Hijos:\n");
while ((entry = readdir(dir)) != NULL) {
    if (es_proceso(entry)) {
        snprintf(path, sizeof(path), "/proc/%s/status", entry->d_name);
        if ((fp = fopen(path, "r")) != NULL) {
            int ppid = 0, pid = atoi(entry->d_name);
            while (fgets(line, sizeof(line), fp)) {
                if (strncmp(line, "PPid:", 5) == 0) {
                    sscanf(line, "PPid: %d", &ppid);
                    break;
                }
            }
            fclose(fp);

            // Imprimir información del proceso padre e hijo
            printf("Proceso PID: %d, PPID: %d\n", pid, ppid);
        }
    }
}

closedir(dir);
}

int main() {
    obtener_memoria();
    listar_procesos();
    return 0;
}
```

Ejecución



The screenshot shows a terminal window titled "Terminal" with the user "jefferson" at the prompt "jefferson@jefferson:~/Escritorio/lab-sopes1". The terminal output displays system memory statistics and a list of running processes with their PIDs and PPIDs.

```
Actividades Terminal 10 de ago 19:01
jefferson@jefferson:~/Escritorio/lab-sopes1

Total RAM: 19900 MB
Free RAM: 15483 MB
Procesos Padre e Hijos:
Proceso PID: 1, PPID: 0
Proceso PID: 2, PPID: 0
Proceso PID: 3, PPID: 2
Proceso PID: 4, PPID: 2
Proceso PID: 5, PPID: 2
Proceso PID: 6, PPID: 2
Proceso PID: 8, PPID: 2
Proceso PID: 9, PPID: 2
Proceso PID: 10, PPID: 2
Proceso PID: 11, PPID: 2
Proceso PID: 12, PPID: 2
Proceso PID: 13, PPID: 2
Proceso PID: 14, PPID: 2
Proceso PID: 15, PPID: 2
Proceso PID: 16, PPID: 2
Proceso PID: 17, PPID: 2
Proceso PID: 18, PPID: 2
Proceso PID: 19, PPID: 2
Proceso PID: 20, PPID: 2
Proceso PID: 21, PPID: 2
Proceso PID: 22, PPID: 2
Proceso PID: 23, PPID: 2
Proceso PID: 25, PPID: 2
Proceso PID: 26, PPID: 2
Proceso PID: 27, PPID: 2
Proceso PID: 28, PPID: 2
Proceso PID: 29, PPID: 2
```