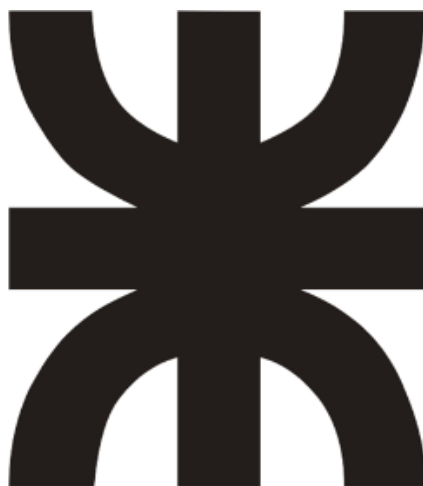


UNIVERSIDAD TECNOLOGICA NACIONAL



Ingeniería en sistemas de información

Asignatura: Sistemas Operativos

Comision: B

Segundo Cuatrimestre

Año: 2023

Trabajo Práctico 2

Tema: Llamadas al sistema

Grupo: 11

Alumnos:

Meyer, Nahuel

nahuelmeyer16@gmail.com

Pividori, Marcos

mpividori3@gmail.com

Simonsini, Juan Pablo

juanpablosimonsini@gmail.com

TABLA DE CONTENIDOS

SECCION PRINCIPAL.....	3
ARGUMENTOS POR LINEA DE COMANDO(ARGC-ARGV-OPCIONES).....	3
Ejercicio 1:.....	3
GESTION DE PROCESOS.....	6
Ejercicio 2:.....	6
Ejercicio 3:.....	10
Ejercicio 3 Factorial:.....	11
GESTION DE SEÑALES.....	14
Ejercicio 4 Tostadora:.....	14
Ejercicio 4 Huespedes:.....	15
Ejercicio 5:.....	16
Ejercicio 6:.....	18
SECCION DE DESCARGOS.....	19
BIBLIOGRAFIA.....	19

SECCION PRINCIPAL

ARGUMENTOS POR LINEA DE COMANDO(ARGC-ARGV-OPCIONES)

Ejercicio 1:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

main(int argc, char *argv[]) {

    int val, activate = 0;

    while((val = getopt (argc, argv, "cls:")) != -1) {
        switch(val){
            case 'c': {
                printf("Cantidad de argumentos: %d", argc);
                printf("\n");
            } break;
            case 'l': {
                int i;
                for(i = 0; i < argc; i++) {
                    printf("Argumento: %d- %s\n", i, argv[i]);
                }
            } break;
            case '?':
                if (optopt == 's') printf(argv[0], ": option requires an argument -- s");
                break;
        }
    }
}
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
        case 's': {
            char *nombreArchivo = optarg;
            FILE *arch = fopen(nombreArchivo, "w");
            if(arch == NULL) {
                printf("El archivo no se creo\n");
                exit(1);
            }
            int i;
            for(i = 0; i < argc; i++) {
                fputs(argv[i], arch);
                fputs("\n", arch);
            }
            fclose(arch);
        }
    }

    activate = 1;

}

if(activate == 0)
    printf("Uso: %s [-cl] [-s nombre_archivo]\n", argv[0]);

return 0;
}
```

```
so2011:~/Desktop# ./ejercicio1
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====
Uso: ./ejercicio1 [-cl] [-s nombre_archivo]
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
so2011:~/Desktop# ./ejercicio1 -c
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====
Cantidad de argumentos: 2
```

```
so2011:~/Desktop# ./ejercicio1 -ls
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====
Argumento: 0- ./ejercicio1
Argumento: 1- -ls
./ejercicio1: option requires an argument -- 's'
```

```
./ejercicio1so2011:~/Desktop# ./ejercicio1 -lc
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====
Argumento: 0- ./ejercicio1
Argumento: 1- -lc
Cantidad de argumentos: 2
```

```
so2011:~/Desktop# ./ejercicio1 -cls salidaejercicio1
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====
Cantidad de argumentos: 3
Argumento: 0- ./ejercicio1
Argumento: 1- -cls
Argumento: 2- salidaejercicio1
```

GESTION DE PROCESOS

Ejercicio 2:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

void imprimir_valor(char *nombre_nodo, int valor, int id_proceso, int id_padre, char *nombre_padre)
{
    printf("%s - VALOR = %d\n", nombre_nodo, valor);
    printf("ID %s: %d - ID padre %s(%s): %d\n", nombre_nodo, id_proceso, nombre_nodo,
nombre_padre, id_padre);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Uso: %s <VALOR>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    int valor = atoi(argv[1]);

    printf("NODO R - VALOR = %d\n", valor);
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
printf("ID proceso raiz: %d\n", getpid());

// Nodo 1
pid_t pid_nodo_1 = fork();

if (pid_nodo_1 == 0) {
    // Proceso hijo Nodo 1
    valor += 100;
    imprimir_valor("NODO 1", valor, getpid(), getppid(), "NODO R");

    // Nodo 1.1
    pid_t pid_nodo_1_1 = fork();

    if (pid_nodo_1_1 == 0) {
        // Proceso hijo Nodo 1.1
        valor *= 2;
        imprimir_valor("NODO 1.1", valor, getpid(), getppid(), "NODO 1");
        exit(EXIT_SUCCESS);
    } else {
        // Proceso padre Nodo 1
        wait(NULL);
    }
}
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
// Nodo 1.2

pid_t pid_nodo_1_2 = fork();

if (pid_nodo_1_2 == 0) {
    // Proceso hijo Nodo 1.2
    valor /= 2;
    imprimir_valor("NODO 1.2", valor, getpid(), getppid(), "NODO 1");
    exit(EXIT_SUCCESS);
} else {
    // Proceso padre Nodo 1
    wait(NULL);
    exit(EXIT_SUCCESS);
}
} else {
    // Proceso padre Nodo R
    wait(NULL);

// Nodo 2

pid_t pid_nodo_2 = fork();

if (pid_nodo_2 == 0) {
    // Proceso hijo Nodo 2
    valor -= 100;
    imprimir_valor("NODO 2", valor, getpid(), getppid(), "NODO R");
```


SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
        exit(EXIT_SUCCESS);
    } else {
        // Proceso padre Nodo R
        wait(NULL);
        exit(EXIT_SUCCESS);
    }
}

return 0;
}
```

```
so2011:~/Desktop# ./ejercicio2 345
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====

NODO R - VALOR = 345
ID proceso raiz: 3224
NODO 1 - VALOR = 445
ID NODO 1: 3225 - ID padre NODO 1(NODO R): 3224
NODO 1.1 - VALOR = 890
ID NODO 1.1: 3226 - ID padre NODO 1.1(NODO 1): 3225
NODO 1.2 - VALOR = 222
ID NODO 1.2: 3227 - ID padre NODO 1.2(NODO 1): 3225
NODO 2 - VALOR = 245
ID NODO 2: 3228 - ID padre NODO 2(NODO R): 3224
```

Ejercicio 3:

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>


int main(int argc, char *argv[]) {
    printf("Id proceso que invoca: %d\n", getpid());

    if (argc != 2) {
        printf("Error. Usar %s [numero]\n", argv[0]);
        printf("Id de Prg-Factorial: %d\n", getpid());
        return 1;
    }

    int number = atoi(argv[1]);

    if (number < 1 || number > 10) {

        printf("Numero fuera de rango: %d\n", number);
        printf("Id de Prg-Factorial: %d\n", getpid());
        return 1;
    }

    char numberStr[12];
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
snprintf(numberStr, sizeof(numberStr), "%d", number);

char *programName = "./ejercicio3_factorial";
char *programArgs[] = {programName, numberStr, NULL};

if (fork() == 0) {
    // Child process
    execv(programName, programArgs);
} else {
    wait(NULL);
}

return 0;
}
```

Ejercicio 3 Factorial:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdlib.h>

int factorial(int n) {
    if (n <= 1) {
        return 1;
    }
}
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
    } else {  
        return n * factorial(n - 1);  
    }  
}  
  
int main(int argc, char *argv[]) {  
    if (argc != 2) {  
        printf("Error. Usar %s [numero]\n", argv[0]);  
        printf("Id de Prg-Factorial: %d\n", getpid());  
        return 1;  
    }  
  
    int number = atoi(argv[1]);  
  
    if (number < 1 || number > 10) {  
        printf("Numero fuera de rango: %d\n", number);  
        printf("Id de Prg-Factorial: %d\n", getpid());  
        return 1;  
    }  
  
    printf("Factorial de %d = %d\n", number, factorial(number));  
    printf("Id de Prg-Factorial: %d\n", getpid());  
  
    return 0;  
}
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
so2011:~/Desktop# ./ejercicio3 367
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pivadori Marcos - mpivadori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====

Id proceso que invoca: 3236
Numero fuera de rango: 367
Id de Prg-Factorial: 3236
```

```
so2011:~/Desktop# ./ejercicio3 3
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pivadori Marcos - mpivadori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====

Id proceso que invoca: 3264
Factorial de 3 = 6
Id de Prg-Factorial: 3265
```

```
so2011:~/Desktop# ./ejercicio3 10
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pivadori Marcos - mpivadori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====

Id proceso que invoca: 3273
Factorial de 10 = 3628800
Id de Prg-Factorial: 3274
```

```
so2011:~/Desktop# ./ejercicio3
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pivadori Marcos - mpivadori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====

Id proceso que invoca: 3275
Error. Usar ./ejercicio3 [numero]
Id de Prg-Factorial: 3275
```

GESTION DE SEÑALES

Ejercicio 4 Tostadora:

```
#include <stdio.h>

#include<signal.h>

#include <unistd.h>

int tostadas=0;

void tostar_negro(){
    tostadas++;
    printf("Deseo tostadas de pan negro.\n");
    printf("Tostadas de pan negro. (%d",tostadas);
    printf(").\n");
    printf("Tostadas listas.\n");
    printf("Tostadora libre,esperando pan.\n");
}

void tostar_blanco(){
    tostadas++;
    printf("Deseo tostadas de pan blanco.\n");
    printf("Tostadas de pan blanco. (%d",tostadas);
    printf(").\n");
    printf("Tostadas listas.\n");
    printf("Tostadora libre,esperando pan.\n");
}
```

```
int main(int argc,char* argv[]){

    signal(SIGUSR1,&tostar_blanco);
    signal(SIGUSR2,&tostar_negro);
    while(tostadas!=30);

    return 0;
}
```

Ejercicio 4 Huespedes:

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc,char* argv[]){

    int tostadora,cont;

    tostadora=atoi(argv[1]);

    for(cont=1;cont<31;cont++){
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
    if(cont%5==0){
        printf("Pedido de Pan Negro. Huesped: %d\n",cont);
        kill(tostadora,SIGUSR2);
        sleep(1);
    }
    else
    {
        printf("Pedido de Pan Blanco. Huesped: %d\n",cont);
        kill(tostadora,SIGUSR1);
        sleep(2);
    }
}
return 0;
}
```

Ejercicio 5:

```
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <signal.h>
```

```
void iterar(){
    int i;
    for(i=0; i<=100000; i++){
```


SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
        if(i%1000==0 && i!=0){
            printf("Iteracion %d \n", i);
        }
        usleep(1);
    }
}

int main(int argc, char *argv[])
{
    signal(SIGINT,SIG_IGN);
    printf("Iteracion INICIADA. Presionar Ctrl-C NO tiene efecto.... \n");
    iterar();
    printf("Computation is done. \n");
    printf("\n");
    printf("REINICIO de la Iteracion. Presionar Ctrl-C AHORA tiene efecto... \n");
    signal(SIGINT,SIG_DFL);
    iterar();
    printf("Computation is done. \n");

    return(0);
}
```

INFORMACION DEL SISTEMA

Ejercicio 6:

```
#include <stdio.h>

#include <sys/utsname.h>

int main() {

    struct utsname systemInfo;

    if (uname(&systemInfo) != 0) {
        perror("uname");
        return 1;
    }

    printf("Tipo de Sistema: %s\n", systemInfo.sysname);
    printf("Nombre del Equipo: %s\n", systemInfo.nodename);
    printf("Version del Kernel: %s\n", systemInfo.release);
    printf("Version del S.O.: %s\n", systemInfo.version);
    printf("Arquitectura: %s\n", systemInfo.machine);

    return 0;
}
```

SISTEMAS OPERATIVOS

TP 2: LLAMADAS AL SISTEMA

```
so2011:~/Desktop# ./ejercicio6
=====
Grupo 11
Meyer Nahuel - nahuelmeyer16@gmail.com
Pividori Marcos - mpividori3@gmail.com
Simonsini Juan Pablo - juanpablosimonsini@gmail.com
=====

Tipo de Sistema: Linux
Nombre del Equipo: so2011
Version del Kernel: 2.6.30-2-686
Version del S.O.: #1 SMP Sat Sep 26 01:16:22 UTC 2009
Arquitectura: i686
```

SECCION DE DESCARGOS

BIBLIOGRAFIA

<https://www.geeksforgeeks.org/getopt-function-in-c-to-parse-command-line-arguments/>

<https://opensource.com/article/21/8/short-option-parsing-c>