07/10/2024 - Laboratorio de SO (Vde@)

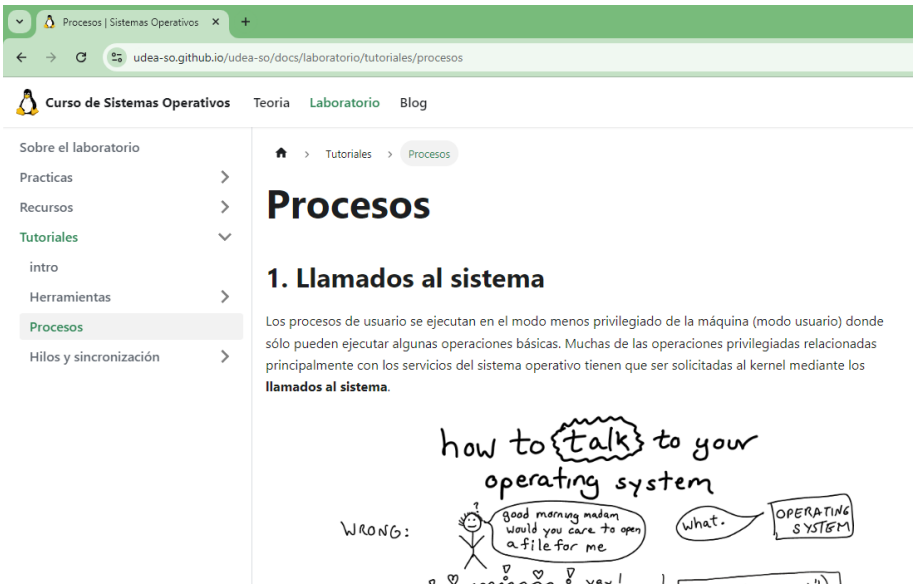
1. Cronograma:

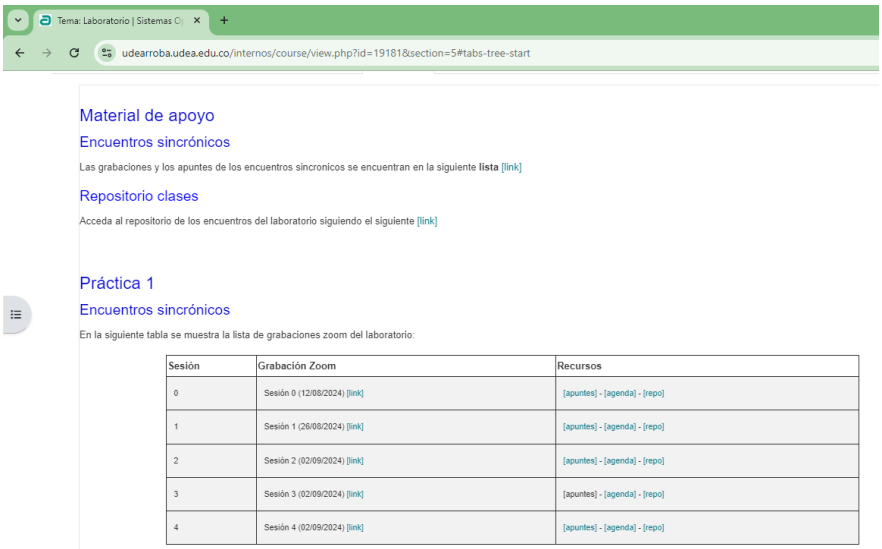
Due time: Dead time:

- Entrega: 28/10/2024 04/11/2024
- Sobre el curso de RHA: No se dejen colgar

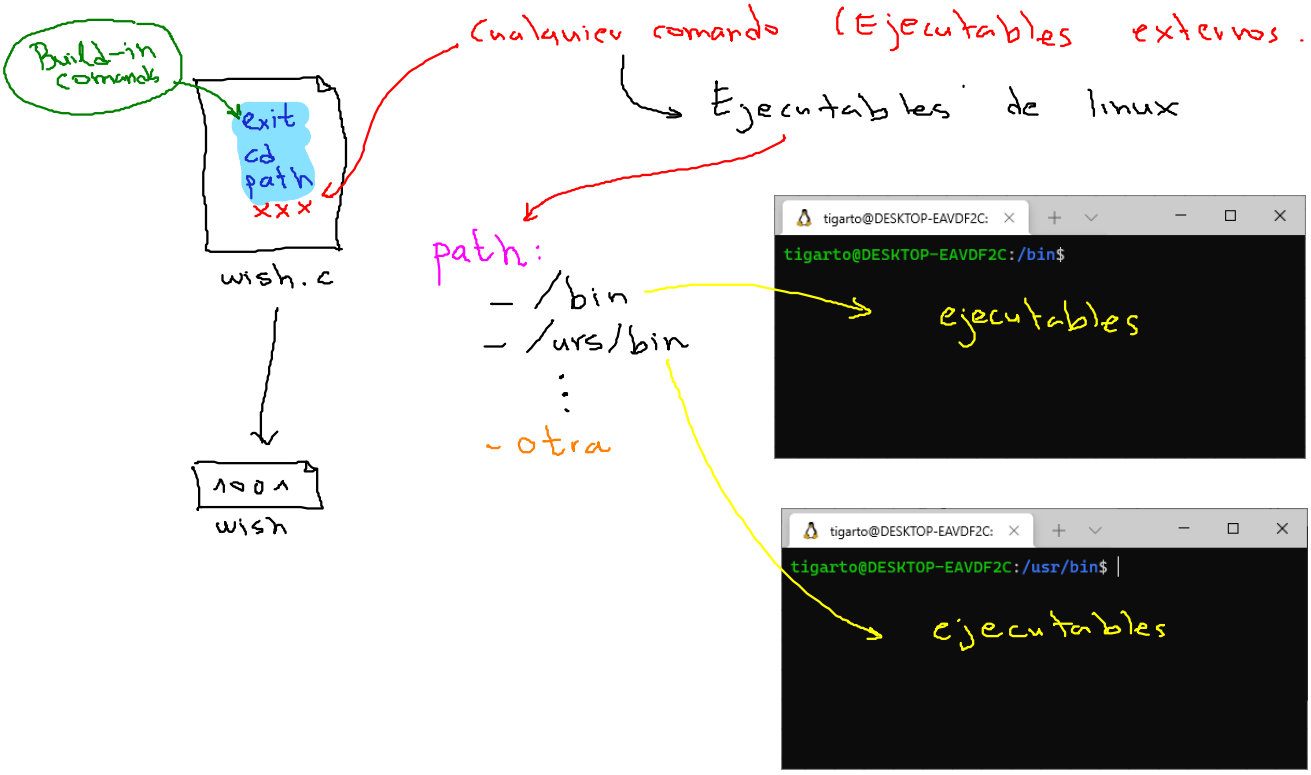
2. Recursos para trabajar:

https://udea-so.github.io/udea-so/docs/laboratorio/intro-laboratorio





3. Aspectos básicos

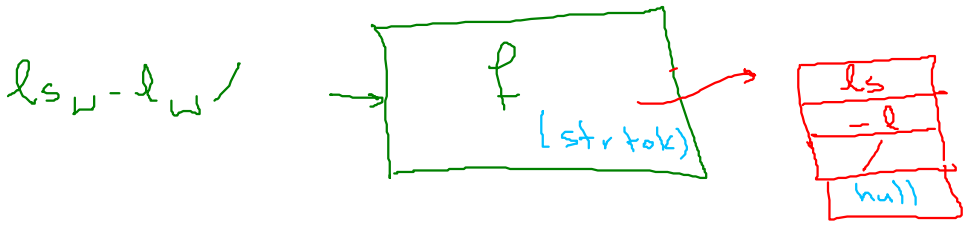


i. Modos de funcionamiento

- a. bash : Como los scripts. Ejemplo:
  - * sh ejemplo.sh
  - * python ejemplo.py
- b. Interactivo: Se entran comandos por teclado

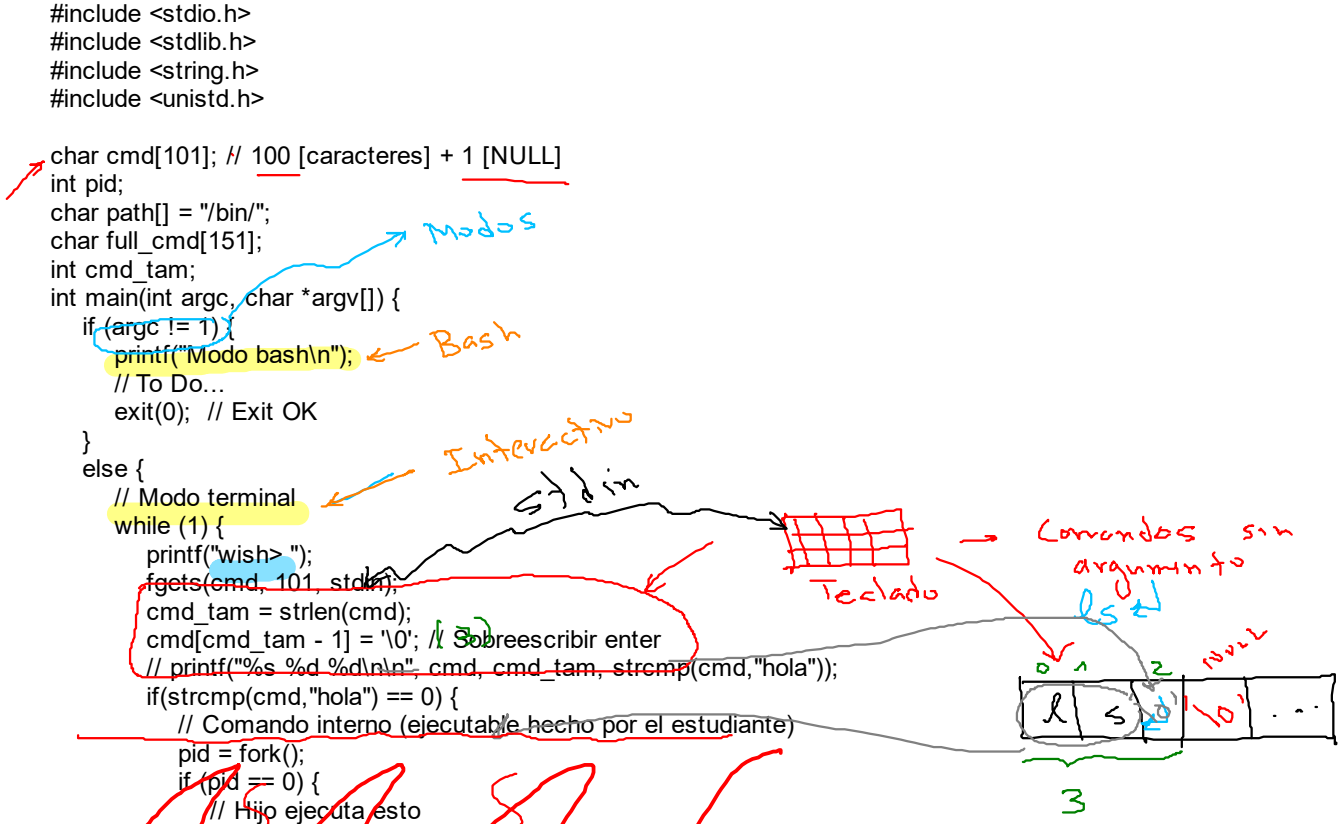
ii. Comandos:

- strsep ~~> strtok ✓



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

char cmd[101]; // 100 [caracteres] + 1 [NULL]
int pid;
char path[] = "/bin/";
char full_cmd[151];
int cmd_tam;
int main(int argc, char *argv[]) {
    if (argc != 1) {
        printf("Modo bash\n");
        // To Do...
        exit(0);  // Exit OK
    }
    else {
        // Modo terminal
        while (1) {
            printf("wish> ");
            fgets(cmd, 101, stdin);
            cmd_tam = strlen(cmd);
            cmd[cmd_tam - 1] = '\0'; // Sobreescribir enter
            // printf("%s %d %d\n\n", cmd, cmd_tam, strcmp(cmd,"hola"));
            if(strcmp(cmd,"hola") == 0) {
                // Comando interno (ejecutable hecho por el estudiante)
                pid = fork();
                if (pid == 0) {
                    // Hijo ejecuta esto
                    execl("./hola", "hola", NULL);
                }
                else {
                    // Padre ejecuta esto
                    wait(NULL); // Padre espera que el hijo acabe
                }
            }
            else if(strcmp(cmd,"exit") == 0) {
                // Comando interno (Se implementa como en el caso de hola)
                exit(0); // Exit OK
            }
            else if(strcmp(cmd,"cd") == 0) {
                // cd -> chdir()
                // Comando interno
                if(fork() == 0) {
                    // Hijo
                    execl("./cd", "cd", NULL);
                }
                else {
                    wait(NULL); // Padre espera que el hijo acabe
                }
            }
            else {
                // Comando externo (De los disponibles en linux - Puede ser el que sea)
                pid = fork();
                switch (pid) {
                case 0:
                    // Hijo
                    strcat(full_cmd, path);
                    strcat(full_cmd,cmd);
                    execl(full_cmd, cmd, NULL);
                    break;
                default:
                    // Padre
                    wait(NULL); // Padre espera que el hijo acabe
                    break;
                }
            }
            // Vaciar el búfer para evitar problemas con el siguiente fgets
            while (getchar() != '\n');
        }
    }
}
```



iii. Tipo de comandos:

→ Build-in Comandos (Internos)

```c
else if(strcmp(cmd,"exit") == 0) {
    // Comando interno
    exit(0); // Exit OK
}
else if(strcmp(cmd,"cd") == 0) {
    // Interno
    // cd -> chdir()
    printf("Uso del cd...\n");
}
```

cd /home/xxx/...

ruta

To do...

→ Externos

```c
if(strcmp(cmd,"hola") == 0) {
    // Comando interno (ejecutable hecho por el estudiante)
    pid = fork();
    if (pid == 0) {
        // Hijo ejecuta esto
        execl("./hola", "hola", NULL);
    }
    else {
        // Padre ejecuta esto
        wait(NULL); // Padre espera que el hijo acabe
    }
```

hijo

padre

```c
else {
    // Comando externo (De los disponibles en linux - Puede ser el que sea)
    pid = fork();
    switch (pid) {
    case 0:
        // Hijo
        strcat(full_cmd, path);
        strcat(full_cmd,cmd);
        execl(full_cmd, cmd, NULL);
        break;
    default:
        // Padre
        wait(NULL); // Padre espera que el hijo acabe
        break;
    }
}
```

/bin

wish> ls

full_cmd

"/bin/"

"/bin/ls"

exec