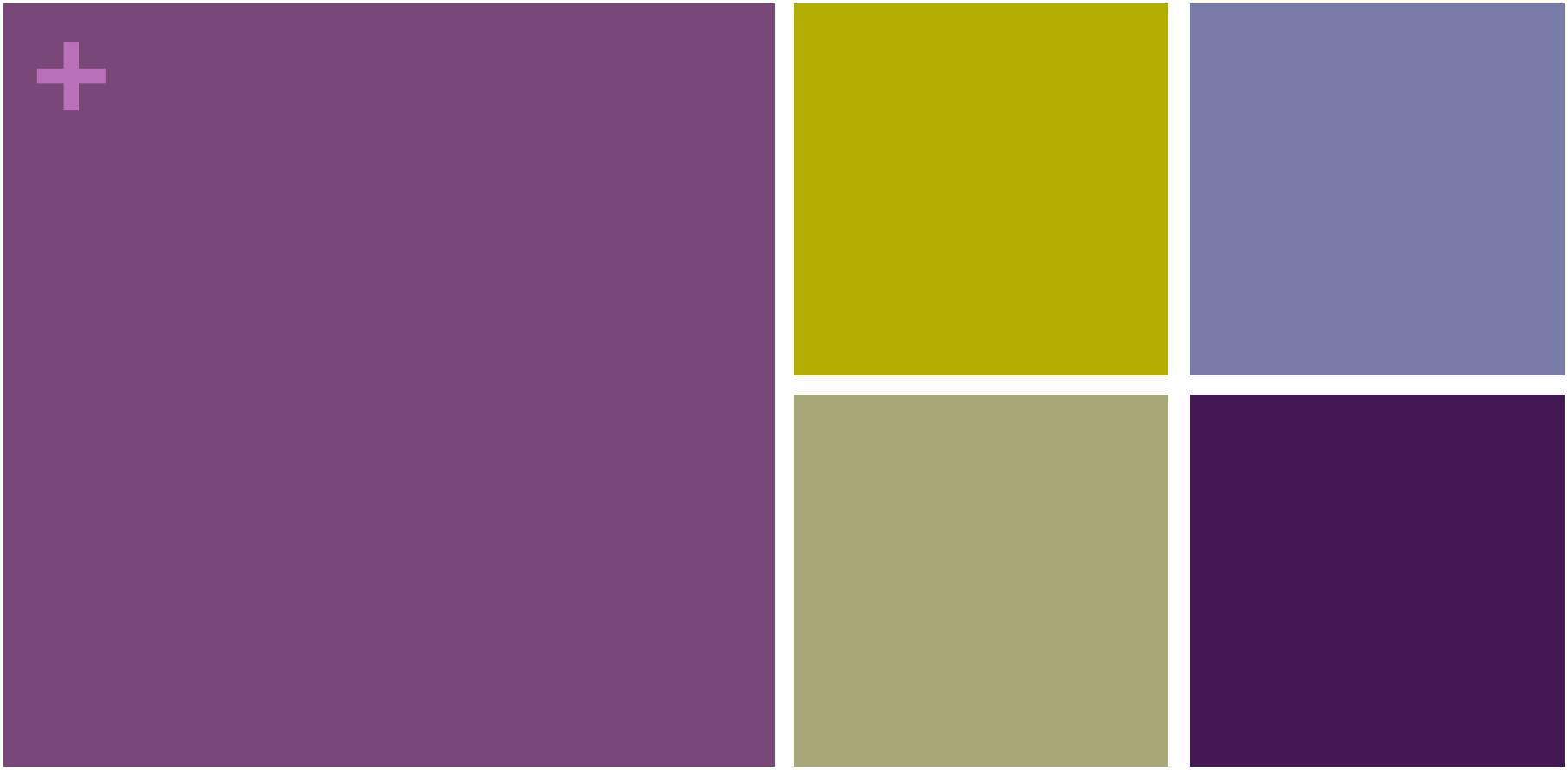

Introducción a GNU/Linux

Procesos, redireccionamientos y tuberías

L.I. Eduardo Iván Ortega Alarcón

Procesos, definición

- Programa
  - Archivo ejecutable que reside en el sistema de archivos.
  - Serie de instrucciones que van siendo ejecutadas en orden.

- Proceso
  - Instancia de un programa en ejecución.

Metáfora de Unix

- Un proceso puede estar:
  - Vivo.
  - Muerto.
  - Puede ser generado.
  - Puede matarse.
- Pueden convertirse en:
  - Zombies.
  - Huérfanos.
    - Lo que implica que un proceso tiene un padre.

PID y PPID

- Cada proceso tiene un ID de proceso, identificado por un número entero, para poder ser reconocido en el sistema como único.
- También los procesos tienen un identificador de su padre PPID.

Creación de un proceso

- Por medio de la llamada al sistema fork(), se crea un proceso.
- El proceso padre realiza la llamada al sistema y crea un hijo.
- Se añade el proceso a la tabla de procesos.

Procesos en primer plano

- Un proceso en primer plano es aquel que toma la terminal en donde se ejecuta.
- La terminal no se puede utilizar hasta que el proceso termine.
- Si se quiere suspender el proceso, es necesario mandar una señal de suspensión.

Procesos en segundo plano

- Un proceso en segundo plano es aquel que se ejecuta de manera "silenciosa", es decir, se sigue ejecutando mientras la terminal está disponible para su uso.
- También se les llama jobs.
- Para ejecutar un comando en segundo plano, es necesario colocar el caracter "&" al final de la línea de comandos.

Tabla de procesos

- Es un arreglo de estructuras con todos los atributos que tiene un proceso, tales como:
  - PID.
  - Prioridad.
  - Estado.
  - Registros de la CPU.
  - Apuntadores a la memoria.
  - Tiempo consumido.
  - Usuario.
  - Etc.

Tabla de procesos

- ps
  - Permite mostrar la tabla de procesos en un formato específico.
    - `$ ps [opciones]`

    - Opciones:
      - -a Procesos de todos los usuarios.
      - -u Formato de usuario.
      - -x Muestra los procesos que tienen terminal asociada.
      - -o Formato personalizado.

Jobs

- Son los procesos en segundo plano.
- Se referencían utilizando el caracter “%”.
- jobs
  - Muestra los jobs que corren en ese momento.
    - `$ jobs`

Reanudar un comando suspendido

- bg
  - Permite reanudar el comando suspendido en segundo plano (background).

    ```
    $ bg %job
    ```

- fg
  - Permite pasar a primer plano un job.
  - Permite reanudar el comando suspendido en primer plano.

    ```
    $ fg %job
    ```

Señales

- Los procesos están a la espera de señales enviadas por el sistema para reaccionar ante un evento.

ID	#	Señal	Descripción
SIGKILL	9	Kill	Muerte. Mata un proceso.
SIGTERM	15	Termination	Indica a un proceso que debe terminar su ejecución.
SIGSTOP	19	Stop	Suspende la ejecución de un proceso.
SIGCONT	18	Continue	Reanuda la ejecución de un proceso con señal SIGSTOP

Enviar señales

- kill
  - El comando kill **sirve para enviar señales** a los procesos.
  - Predeterminadamente envía la señal 15 (SIGTERM).

```
$ kill [-señal] pid1 … pidN
```

Redireccionamientos y tuberías

Descriptores de archivos

- El kernel mantiene una tabla de archivos abiertos para cada proceso que se ejecuta.
- En ésta tabla se definen los archivos abiertos que hay en el sistema y se les asigna un descriptor de archivos (fd).
- Un fd es un número entero positivo.
- Cuando se realiza la llamada al sistema open(), devuelve el fd.

Entrada y salida estándar

- Cuando se crea un proceso, el kernel define qué archivos son los que necesitará el proceso para leer y escribir.
- Cada proceso en el sistema operativo tiene asignados tres archivos abiertos.
  - `STDIN`
    - Con fd = 0
  - `STDOUT`
    - Con fd = 1
  - `STDERR`
    - Con fd =2

Redireccionamientos

- Existen 4 operadores básicos para redireccionar la entrada, salida o error estándar.

>	Redirecciona la salida de forma destructiva.
>>	Redirecciona la salida de forma no destructiva (añade).
<	Redirecciona la entrada a partir de un archivo.
<<	Redirecciona la entrada del teclado hasta una cadena específica

Redireccionamientos

Redireccionamiento	Sintaxis
STDOUT a archivo	`$ cmd > archivo`
STDERR a archivo	`$ cmd 2> archivo`
STDOUT y STDERR a archivo	`$ cmd > archivo 2>&1`
STDIN de archivo	`$ cmd < archivo`
STDIN de teclado hasta cadena	`$ cmd << cadena`
Añadir STDOUT a archivo	`$ cmd >> archivo`
Añadir STDERR a archivo	`$ cmd 2>> archivo`
Añadir STDOUT y STDERR a archivo	`$ cmd >> archivo 2>&1`

Tuberías

- Una tubería funciona para enviar la salida o error estándar como entrada de otro comando.
- Se utiliza el caracter | (pipe).
- Ejemplos:

Redireccionamiento	Sintaxis
STDOUT de cmd a STDIN de cmd2	`$ cmd | cmd2`
STDOUT y STDERR de cmd a STDIN de cmd2	`$ cmd 2>&1| cmd2`