

Guía de Trabajos Prácticos

Esta guía contiene los Trabajos Prácticos (TP) obligatorios del curso R1042, incluyendo una guía para su entrega con el formato de presentación y un cronograma con las fechas de entrega.

Durante el transcurso del año se deberán entregar Trabajos Prácticos, que se caracterizan por ser incrementales. Esto significa que en cada TP se irán introduciendo nuevos conocimientos con respecto al TP anterior. A su vez, se deberán utilizar las herramientas SVN, Makefile y Doxygen cuando se lo indique.

Formato de presentación

- Los archivos fuentes deben tener en todos los casos los comentarios necesarios para clarificar su lectura.
- Cada subrutina/función, debe contar con un encabezado describiendo la operación que realiza, los parámetros que espera como entrada, y los resultados que debe presentar, indicando formato y método de entrega.
- Como encabezado del programa, debe haber un comentario que explique claramente que hace dicho programa, y las instrucciones detalladas (comandos) para su compilación y “linkeo”.
- TODOS los ejercicios son obligatorios.
- **La entrega de TODOS los trabajos prácticos es obligatoria para regularizar la materia.**

Trabajo Práctico N°9

- Archivos -

Ejercicio 1

Escriba un programa que lea el archivo de texto "TTT.TXT" y determine:

- Cantidad total de palabras
- Cantidad de veces que aparece la palabra "diodo"
- Escribir una función que permita buscar cualquier palabra e indique cuántas veces la encontró.

Nota: El archivo TTT.TXT deberá contener el siguiente párrafo:

Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido.¹ Este término generalmente se usa para referirse al diodo semiconductor, el más común en la actualidad; consta de una pieza de cristal semiconductor conectada a dos terminales eléctricos. El diodo de vacío (que actualmente ya no se usa, excepto para tecnologías de alta potencia) es un tubo de vacío con dos electrodos: una lámina como ánodo, y un cátodo. De forma simplificada, la curva característica de un diodo (I-V) consta de dos regiones: por debajo de cierta diferencia de potencial, se comporta como un circuito abierto (no conduce), y por encima de ella como un circuito cerrado con una resistencia eléctrica muy pequeña. Debido a este comportamiento, se les suele denominar rectificadores, ya que son dispositivos capaces de suprimir la parte negativa de cualquier señal, como paso inicial para convertir una corriente alterna en corriente continua.

Ejercicio 2

El archivo binario BIPOLAR.DAT contiene una rudimentaria base de datos de transistores bipolares. La base tiene registros de cinco campos:

- Nombre del transistor: alfabético – 10 caracteres
 - Tipo: alfabético de tres caracteres
 - hFE: valores enteros entre 50 y 420
 - Potencia: valores reales entre 10mW y 12W
 - Tensión de ruptura: valores enteros entre -130V y +130V
- Se pide escribir un programa que permita la carga y actualización del archivo.
 - Escribir un programa que ingresando el nombre del transistor, muestre por pantalla los datos correspondientes al mismo.

- c. Escribir un programa que dado el hFE informe que modelos de transistores poseen ese valor ese valor.

Nota: El archivo BIPOLAR.DAT deberá contener los siguientes datos al abrirse por primera vez:

BC548,TBJ,120,0.5,30

BD138,TBJ,250,12.5,60

BC327,TBJ,600,0.600,-45

BC337,TBJ,500,0.625,45

- Nombre del transistor: alfabético – 10 caracteres
- Tipo: alfabético de tres caracteres
- hFE: valores enteros entre 50 y 620
- Potencia: valores reales entre 10mW y 15W
- Tensión de ruptura: valores enteros entre -130V y +130V

Ejercicio 3

Se desea disponer de una función para depurar un programa, volcando la información sobre un archivo de texto histórico (log file), de acuerdo al siguiente prototipo:

status LogDebug(char *archivo, char* datos);

en donde “archivo” contiene la ruta al archivo sobre el cual añadir los datos (sin borrar los ya existentes), “datos” es la cadena de caracteres con la información a guardar, y “status” un tipo enumerativo con los símbolos ERROR y OK. Se pide:

- a) Definir los tipos status y string.
- b) Escribir el código de la función, teniendo en cuenta que el puntero “datos” puede ser nulo, producto de información faltante.
- c) Dar un ejemplo de invocación de la función.

Ejercicio 4

Escribir un programa que concatena el contenido de dos archivos de texto cuyos nombres recibe desde la línea de comandos. El archivo final se llamará *resultado*