A Contexto

Para llevar a cabo su propósito, buscadores como Google o Yahoo deben de indexar enormes cantidades de páginas Web. De acuerdo con el portal WorldWideWebSize.com (http://www.worldwidewebsize.com/) se han llegado a indexar cerca de 60 mil millones de páginas en la Web. Por tanto, además de precisar una importante infraestructura física y un costoso soporte hardware, se hace imprescindible el empleo de mecanismos muy eficientes para el almacenamiento de la información y el diseño de algoritmos extremadamente rápidos para llevar a cabo los procesos de búsqueda sobre esa ingente cantidad de información.

El objetivo de esta práctica es crear un buscador de páginas Web que haga uso de una técnica apropiada para almacenar el contenido de las páginas y que permita encontrar de manera eficiente las páginas que incluyen ciertas palabras, ya sean algunas de ellas (búsqueda con OR) o todas ellas (búsqueda con AND).

B El Problema

Analizar, diseñar e implementar un buscador de páginas Web por contenido. El programa admitirá una serie de comandos, que se leerán siempre de la entrada estándar, produciendo el resultado en la salida estándar. Los comandos admisibles son los siguientes:

- **Insertar nueva página**: indicando la URL, título, relevancia y contenido en sí de la página. El formato de entrada se describe más adelante.
- **Buscar palabras con AND**: listar todas las páginas donde aparecen ***todas*** las palabras dadas. Las palabras se definen como cualquier sucesión de una o más letras delimitadas por caracteres que no sean letras.
- **Buscar palabras con OR**: listar todas las páginas donde aparezcan ***algunas*** de las palabras dadas.

El programa diseñado deberá cumplir los siguientes requisitos:

- Las búsquedas deben ser independientes de mayúsculas/minúsculas y de las tildes.
- Se requiere que no se pierda ni una sola búsqueda para aprobar la práctica.
- El programa tiene que estar bien diseñado para conseguir la máxima eficiencia de tiempo y de memoria (con especial hincapié en lo primero). Por ejemplo, no será admisible usar una simple lista en la que almacenar y recorrer todas las páginas para cada búsqueda nueva.

C Formato de entrada

La entrada está compuesta de varios comandos. Todos los comandos están en una línea distinta, excepto la inserción, que ocupará varias. Los comandos admisibles son: **i** (para insertar), **a** (búsqueda con AND), **o** (búsqueda con OR), **s** (salir). La entrada acabará siempre con un comando **s**. El formato de los comandos es el siguiente:

- **Insertar**: después de la **i** aparecerá un entero R que indica la relevancia que se le asigna a la página (suponer un número natural). La siguiente línea contiene la URL de la página (que será única para cada página), y la siguiente el título de la misma. A continuación viene el contenido de la página en sí. Cada página puede ocupar una o varias líneas, y acaba con la palabra clave "FinDePagina", que es independiente de mayúsculas, minúsculas o tildes. Este puede ser, por ejemplo, un comando **i** válido:

  ```
  i 5
  http://www.um.es
  Universidad de Murcia
  La Universidad de Murcia del tercer milenio tiene como eje central de su actividad la consecución de la excelencia académica y científica.
  Todo ello queda recogido en esta Web que pretende satisfacer las necesidades no sólo de la comunidad universitaria sino de la Sociedad en general.
  FINDEPAGINA
  ```

- **Búsqueda con AND**: después de la letra **a**, aparecerá una lista de una o más palabras que se buscan. Por ejemplo:

  ```
  a llevo la Condición
  a la PROGRAMACION de Meyer
  ```

- **Búsqueda con OR**: después de la letra **o**, aparecerá una lista de una o más palabras que se buscan. Por ejemplo:

  ```
  o condicion
  o TROYANO TROYA
  ```

D Formato de salida

Después de cada comando, se mostrará por pantalla información sobre el resultado del mismo. La salida tendrá el siguiente formato:

- **Insertar**: la salida serán 2 líneas, que contendrán: (1) el número ***consecutivo*** de la página respecto al resto de páginas introducidas, la URL, el título y la relevancia de la página insertada (separados por comas), y (2) el número total de palabras leídas, ya sean repetidas o no. Por ejemplo, la salida para el ejemplo de arriba será:

  ```
  1. http://www.um.es, Universidad de Murcia, Rel. 5
  46 palabras
  ```

- **Búsqueda con AND y con OR**: los resultados irán numerados de forma consecutiva, y para cada uno se indicará: la URL, el título y la relevancia, separados por comas. La lista estará ordenada de acuerdo a la relevancia de la página (de mayor a menor), y en caso de empate por título, alfabéticamente. Por ejemplo, el resultado para la búsqueda "o actividad" en el ejemplo de arriba sería:

  ```
  1. http://www.um.es, Universidad de Murcia, Rel. 5
  Total: 1 resultados
  ```

E Ejercicio opcional

La mayoría de los buscadores Web han evolucionado a lo largo de los años en consonancia con los avances tecnológicos que se han ido produciendo. De un tiempo a esta parte, buscadores como Google incluyen una utilidad de autocompletar palabras que explota los entresijos de la tecnología Ajax. El usuario teclea unas pocas letras que componen la palabra a buscar y espera que el buscador ofrezca varias posibilidades de

búsqueda que comienzan con el mismo prefijo. Evidentemente se requieren mecanismos muy eficientes para que la experiencia del usuario sea fluida.

Este ejercicio adicional llevará a cabo la facilidad de autocompletado. Dado un prefijo, se mostrará una lista de las palabras (de entre las contenidas en las páginas web insertadas) que comparten ese prefijo. El formato de entrada será el siguiente. Aparecerá una letra **p** seguida del prefijo que se busca. En la salida, cada palabra resultante aparecerá en una línea, precedida de un número consecutivo. Al final aparecerá el número total de palabras resultantes. Las palabras estarán en orden alfabético.

Por ejemplo, tras la introducción de la página de ejemplo del apartado D, el comando: "p co" produciría la salida:

1. como
2. comunidad
3. consecucion

Total: 3 resultados

Deberá proporcionarse una implementación eficiente para el almacenamiento de las palabras y la consulta de sus prefijos. Para ello se propone la creación de un árbol trie de caracteres, que se irá "llenando" conforme se lean las palabras que conforman las páginas. La búsqueda para el autocompletado consistirá en recorrer todas las ramas del árbol trie a partir del prefijo común. Este ejercicio se integrará con lo realizado en el resto de la práctica. Su evaluación corresponderá al 20% de la nota de la práctica.

F Fases de desarrollo

Para una correcta resolución de la práctica, se sugiere al alumno que satisfaga la siguiente planificación en los plazos señalados abajo. Esta planificación se seguirá en las sesiones de prácticas presenciales de la asignatura, y se validará en el juez on-line. Su cumplimiento repercutirá de forma positiva en la evaluación de la práctica. Las fases de desarrollo son:

Semana 1. Hasta el 3 de noviembre: resolver el problema de introducción al Mooshak (000) y la mayor cantidad de problemas del concurso "AED 10/11. Seminarios de C y C++".

Semana 2. Hasta el 12 de noviembre: implementar la ordenación de vectores de enteros (027), resolver los problemas básicos referentes a normalizar un texto (028 y 029), separar las palabras (030) y definir el tipo **`Pagina`** (031).

Semana 3. Hasta el 22 de noviembre: implementar el mecanismo de almacenamiento de las páginas y el modo de hacer referencias a las mismas (225) y desarrollar la lista ordenada de referencias (226).

Semana 4. Hasta el 29 de noviembre: hacer uso de un tipo diccionario donde la clave son cadenas y el valor es una lista ordenada de referencias, y crear la operación **`cargarPagina`**, que meta las apariciones de palabras de una página en el diccionario (227), y definir la operación **`buscarAND`** (228).

Semana 5. Hasta el 6 de diciembre: definir la operación **`buscarOR`** (229).

Semana 6. Hasta el 15 de diciembre: completar la práctica (230) y hacer el ejercicio opcional de autocompletar palabras (327). Entrega final de la memoria de la práctica.

En todos los casos, se podrán realizar los envíos al juez on-line hasta las 23:30:00 del día señalado. Los grupos que cumplan todos los plazos señalados y de forma satisfactoria tendrán un +1 en la nota final de la práctica.

G Documentación

La documentación final contendrá los siguientes apartados:

1. **Portada**. Nombre de los alumnos, grupo y subgrupo de prácticas, e-mail de cada uno y cuenta del Mooshak usada.
2. **Análisis del problema y diseño de la solución**. Encontrar los tipos abstractos que aparecen en el problema, y en qué partes aparecen. Analizar las diferentes alternativas que se presentan para la implementación de esos tipos o para la elección de los tipos de la STL de C++. Mostrar un esquema gráfico global de las estructuras de datos existentes. Detallar la descomposición modular del programa. Documentar cualquier otra decisión de diseño que pueda resultar de interés.
3. **Listado de los envíos**. Sacar un listado de todos los envíos realizados, en el que se vean la fecha y hora de los envíos. Para los envíos no aceptados, indicar de forma muy breve la causa del rechazo.
4. **Listado del código**. Incluyendo el fichero `makefile` necesario para compilar. Incluir sólo el listado de la práctica completa.
5. **Informe de desarrollo**. Describir cómo ha sido la coordinación y el reparto del trabajo entre los miembros del grupo. Rellenar las tablas de dedicación personal en las distintas fases del trabajo. Se utilizarán tablas como las explicadas en las páginas 37 y 350 del texto guía, rellenadas con el mayor rigor posible.
6. **Conclusiones y valoraciones personales**.

H Evaluación de la práctica

H.1 Obligatorio

Para aprobar la práctica se requiere que:

- El programa se pueda ***compilar sin errores*** y, para todos los ejercicios, el código debe haber sido aceptado (resultado "*Accepted*") por el juez on-line de la asignatura (Mooshak). En particular, el programa estará escrito en C++, y el código se deberá compilar en Linux.
- El programa debe ***funcionar correctamente***, sin colgarse y produciendo ***resultados correctos*** para el conjunto de pruebas que se determinen. Para ello, el profesor puede usar (pero no está limitado a) los casos de prueba incluidos en el juez on-line de la asignatura.
- La ***memoria de la práctica*** debe contener todos los puntos indicados en el apartado G, y debe ser entregada en el plazo que se establezca. ¡La documentación entregada no debe contener *faltas de ortografía* (incluida la omisión de tildes)!

H.2 Criterios de valoración

La práctica se puntuará de acuerdo con los siguientes criterios de calidad del software:

- **Análisis y diseño**. Se valorará la calidad y adecuación del diseño y el análisis realizados, y la dedicación a estas fases previas a la implementación. Se deben encontrar los tipos abstractos que aparecen, e implementarlos usando clases, eligiendo las estructuras más adecuadas.
- **Modularidad**. La funcionalidad debe estar bien repartida entre los módulos. Debe estar claro el sentido y la responsabilidad de cada módulo. Se debe respetar el principio de ocultación de la implementación.
- **Uso del lenguaje**. El código deber ser claro, legible, robusto y eficiente. No crear procedimientos muy largos y complejos. Se valorará el uso de clases genéricas (plantillas) y precondiciones / postcondiciones (asertos).
- **Seguimiento continuo**. El correcto cumplimiento de las fases de desarrollo, marcadas en el apartado F, será un aspecto a favor de la evaluación de la práctica.
- **Comando opcional**. Recordamos que los comandos obligatorios corresponden a un 80% de la nota de la práctica y el comando opcional al 20% restante.

H.3 Otras cuestiones

La práctica se deberá realizar preferiblemente en **grupos de dos alumnos**. De forma extraordinaria se permiten **grupos de 1 alumno**, pero no se prevé ninguna reducción del trabajo para los mismos.

Para realizar pruebas y para la verificación de las fases de desarrollo, los profesores dejarán en la página Web del juez on-line (http://dis.um.es/~mooshak/), dentro del concurso “AED 10/11. T2 y T3: Conjuntos y Árboles”, los problemas mencionados en el apartado F. Cada alumno dispondrá de un login y password para acceder a este sistema; el grupo deberá elegir y utilizar una de las cuentas para hacer los envíos al juez.

La fecha de entrega definitiva de esta práctica es el **16 de diciembre de 2010**. La forma de hacer las entregas (en papel, por email, a través de SUMA, etc.) lo indicará cada profesor de prácticas.

AVISO IMPORTANTE

Existe un mecanismo de comprobación automática de todos los envíos, de todos los grupos, de todas las titulaciones. La copia de esta práctica (fuera de la coincidencia casual) supondrá el suspenso fulminante de toda la asignatura en la convocatoria que corresponda, para los grupos implicados en la copia, con la consiguiente anulación del resto de actividades de evaluación continua.