

Divide y Vencerás

Allan Raúl Valentín Acuña Otero

Facultad de Ingeniería y Arquitectura, Universidad Americana

Fernando Gabriel Gómez Gazo

Facultad de Ingeniería y Arquitectura, Universidad Americana

Armando José Valdivia Reyes

Facultad de Ingeniería y Arquitectura, Universidad Americana

Guillermo José Vega Vega

Facultad de Ingeniería y Arquitectura, Universidad Americana

Ing. José Durán

SIS0107:: Metodología y Programación Estructurada

Índice

Divide y Vencerás.....	3
Estrategia Universal.....	4
Objetivo.....	4
Beneficios.....	5
Código Espagueti.....	5
Creación de Algoritmos.....	5
Resultados.....	6
Paralelismo.....	7
Desventajas.....	7
Recursos del Sistema.....	7
Complejidad de Relaciones.....	7
Overhead de la Recursividad.....	8
Espacios de Memoria.....	8

Divide y Vencerás

El avance de la tecnología y la creciente dificultad de nuevos problemas piden un desarrollo más fácil de manejar, por eso el paradigma “Divide y vencerás” es algo importante en este contexto, demuestra una herramienta de vital importancia para abordar temas sobre desafíos, trabajos y resolución de problemas. Este paradigma se basa en una forma en la que una idea se puede descomponer en varias partes para así llevar un mejor manejo de esta, y tener una solución a la problemática más sencilla.

La presente investigación se enfoca en profundizar el paradigma “Divide y vencerás”, con un vistazo a una explicación práctica y al impacto de esta en el mundo de la computación, gracias al uso de diversos algoritmos basados en este paradigma y a la documentación científica consultada se busca explicar y mostrar las ventajas y limitaciones de esta perspectiva.

Un paradigma según Canelo (2024) se define como “una manera o estilo de programación de software”, en resumen, son los diversos caminos que tiene un ingeniero para la creación de proyectos informáticos, cada paradigma posee una filosofía única, basada en la aplicación de diversos conceptos e ideas informáticas, como la programación orientada a objetos, de la cual se basa todo un paradigma completo.

Con este antecedente se establece la pregunta ¿cuál es la idea principal del enfoque, divide y vencerás? La idea es sencilla, consiste en dividir una problemática en parte más sencilla para disminuir su complejidad, en esta idea se aplica conceptos como la recursividad, esta es una estrategia en la que la una función se emplea a sí misma para resolver un problema más pequeño,

esto es a lo que comúnmente se le dice que una función se llame a ella misma (Euroinnova International Online Education, 2024).

Estrategia Universal

El nombre de este paradigma es uno muy famoso y propagado a lo largo de la historia, tal es su relevancia que en todas las disciplinas, sin importar la naturaleza de la misma, adoptan de cierta forma este enfoque. El éxito de este se debe principalmente a los resultados obtenidos, los cuales en su mayoría son positivos, esto se debe a la naturaleza de su resolución, esta consiste en dividir una unidad, en este caso problema, en varias partes para poder atacar una a una en busca de generar facilidades.

El origen de esta célebre frase se desconoce, algunos historiadores se la atribuyen al antiguo gobernante romano Julio César, sin embargo, a pesar de esta incertidumbre, se conoce que es una estrategia tan antigua, que fue empleada en las batallas del mundo antiguo. Una guerra es una clara analogía para entender cómo funciona este paradigma, es más fácil atacar un batallón dividido que a todo un ejército junto, pues tanto los recursos como la complejidad del ataque aumentan en comparación a un sector de este. En este caso el enemigo sería el problema a solucionar y los recursos, la lógica, al enfrentarse a un problema menor, la lógica es menos compleja, lo que permite una mayor legibilidad para el programador.

Objetivo

El principal objetivo que se quiere alcanzar con la aplicación de este paradigma es la facilidad de resolución, a medida que la complejidad de los problemas aumenta es necesario de una estrategia eficaz para la resolución de problemas, por eso la planificación es un apartado importante en esta metodología, pues permite al programador prepararse y pensar de qué manera dividir la problemática para lograr una solución óptima. Este método exige a los ingenieros de

sistemas abordar qué realicen un análisis jerarquizado del programa, en el cual se plantea que funciones albergan a otras, lo cual define los niveles de dependencia de un módulo a otro.

Beneficios

Código Espagueti.

A la hora de emplear otros enfoques se puede caer en el error de diseñar un código espagueti, un término empleado para referirse a un segmento de código poco entendible para las personas y el propio programador, esto se debe a que no se define de antemano los pasos ordenados a llevar a cabo. Para evitar que esto ocurra, el paradigma sugiere que se divida cada acción en un módulo, es decir, que se divida el cuerpo del programa en varias ramas y que cada una de esta ejecuta únicamente una acción, lo que mejora la lectura de código.

Cabe destacar, que este paradigma es empleado generalmente por personas que inician en el mundo de la programación, es útil porque enseña la base de la lógica de un algoritmo y disminuye los riesgos de realizar malas prácticas de programación, en la cual se incluye la generación de código espagueti.

Creación de Algoritmos

Uno de los principales beneficios es que este enfoque ha sido la base para la creación de diversos algoritmos de gran utilidad, pues la efectividad de este paradigma ayuda a crear métodos de resolución que ocupan menos recursos y son excelentes para aplicar, tanto en grandes como pequeños proyectos.

Merge Sort. Este algoritmo empleado para ordenar los elementos de una lista dividiendo esta misma hasta que cada sublista tenga solamente un elemento, luego de esto se ordenan de

forma ascendente o descendente, este método disminuye los riesgos totales de que sucedan evaluaciones incorrectas o no compare correctamente.

Quick Sort. Al igual que el anterior, se trata de un método de ordenamiento, el cual se basa en el concepto de un pivote, el cual divide una lista en dos sublistas que se ordenan en torno al pivote, luego se unen ambas sub arreglos para obtener el ordenado.

Búsqueda Binaria. Es un algoritmo que se utiliza para encontrar un determinado elemento en una lista ordenada, en la cual se escoge un número que es la mitad del arreglo, en la que se evalúa si el número a buscar se encuentra en una mitad u otra, luego esa mitad se vuelve a dividir hasta encontrar un número.

Resultados

Una de las principales ventajas de adoptar este paradigma es que crea programas que administran mejor los resultados de los procesos, esto ayuda a que el programador alcance su objetivo. Sin embargo, su máxima ventaja es la fácil comprensión, pues evita ambigüedades, posibles excepciones y un mayor manejo de la metodología aplicada, no hay nada mejor que tener todo el control acerca de lo que hace, porque, a través de este medio es más fácil detectar los errores que produce el programa, esto ayuda a crear soluciones rápidas para nuevos problemas.

La magnitud de la influencia de este sistema en el éxito de los programas, que es considerada su aplicación como una buena práctica de programación, por lo que todo software existente aplica a su manera este lema tan famoso.

Paralelismo

Esa propiedad del paradigma “divide y vencerás” es importante desde el punto de vista resolutivo, al tratarse de procesos aparentemente independientes estos pueden ocurrir de forma simultánea e ininterrumpida, a menos de que mantengan una relación de dependencia, sin embargo, al permitir que una función pueda llamar a cualquier otra, no se restringe la cantidad de usos o dependencias que cada módulo posee.

Desventajas

Recursos del Sistema

Al ser algoritmos y métodos minuciosos, se consume mucha memoria de las computadoras, un problema que limita la portabilidad de los proyectos, debido a que la mayoría de los dispositivos de gama baja no serían capaces de portar estos programas de forma cómoda e interactiva. Por otro lado, al no ser tan eficientes, se podrían emplear otros paradigmas que empleen menos recursos y realicen el mismo trabajo, esta debilidad se evidencia con los algoritmos, por ejemplo, la búsqueda binaria necesita de muchas divisiones para poder encontrar un número, imagina un arreglo gigante de datos, ¿cuántas divisiones debería realizar el sistema para encontrar un número particular?

Complejidad de Relaciones

Los módulos deben tener una relación jerárquica unos con otros, no obstante, estas reglas pueden generar confusiones y malas prácticas, como la dependencia de un módulo de alto nivel

con uno de bajo nivel, ya que provoca confusión y un código espagueti. En resumen, se debe tener cuidado con las implementaciones de los módulos, puesto que las relaciones entre estos pueden ser ambiguas y dañar la legibilidad del código fuente.

Overhead de la Recursividad

Este término se refiere a los costos adicionales de tiempo y recursos que tienen los programas para desempeñar la recursividad, aunque resulte útil en algunas situaciones, esta técnica recarga mucho la memoria y los procesos llevados a cabo, ya que se debe guardar la ejecución de la función actual, luego ejecutar la otra y almacenar su ejecución, para ello se tiene que evaluar si realmente es necesaria su implicación o si hay otros métodos más viables que puedan completar la tarea sin recurrir a esta estrategia de forma enredosa o mal aplicada.

Espacios de Memoria

Al tener que dividir tantas veces una tarea, en ocasiones se asignan espacios de memorias innecesarios o arbitrarios, como los empleados en los métodos Merge Sort o Búsqueda Binaria, en las que surge la necesidad de asignar espacios para subdivisiones que, en otros métodos como el Bubble Sort no se tendrían que ocupar. El que un proyecto ocupe espacios innecesarios no lo hace eficaz ni eficiente, por lo tanto, se tendría que optar por alternativas más adecuadas por el contexto dado.

En conclusión, la perspectiva “Divide y Vencerás”, es un método que ayuda al programador a comprender el problema para su posterior solución, permite crear soluciones sencillas a partir de la división de tareas, una estrategia que siempre resulta ganadora sin importar el medio en el que se aplica. Como programadores, se debe de conocer las dos caras de emplear este recurso algorítmico, por lo que, el conocimiento y la experiencia son factores clave para lograr la correcta aplicación de un paradigma tan completo como este, que, a pesar de parecer sencillo, requiere de una serie de pasos de preparación que guíe al programador a una segura, eficiente e innovadora automatización de procesos, ante este panorama, como ingenieros informáticos ¿Consideran esta estrategia la más viable para la resolución de problemas?

Referencias

- Acuña, A. (2024). Divide y vencerás. Canva.
https://www.canva.com/design/DAGTOCL0-k0/fHLrBUgZnPw_sVZRuGmPXQ/edit?utm_content=DAGTOCL0-k0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
- Canelo, M. M. (2024, 21 marzo). *¿Qué son los paradigmas de programación? Profile Software Services*. <https://profile.es/blog/que-son-los-paradigmas-de-programacion/>
- Demonics. (2016, 11 agosto). *Algoritmos Divide y vencerás*. Aprende y Programa.
<https://aprendeyprogramablog.wordpress.com/2016/08/09/algoritmos-divide-y-venceras/>
- Euroinnova International Online Education. (2024, 17 septiembre). *¿Qué es y cuáles son los ejemplos de conocimiento científico?*
<https://www.euroinnova.com/programacion-y-desarrollo-de-software/articulos/recursividad-en-informatica>
- Khan Academy. (s. f.).
<https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search>
- Programación II - Algoritmo MergeSort. (s. f.).
<https://sites.google.com/site/programacioniiiuno/temario/unidad-6---anlisis-de-algoritmos/algoritmo-mergesort>
- Allnrvao. (s. f.). *GitHub - allnrvao/Investigacion_Divide_Venceras: Repositorio de investigación, divide y vencerás*. GitHub.
https://github.com/allnrvao/Investigacion_Divide_Venceras.git