



Práctica 3: Preguntas y Bonus

Grupo: GPT-Masters

Profesor: Carlos Jaime Barrios Hernández

Escuela de Ingeniería de Sistemas e Informática

Arquitectura de computadores - A2

Universidad Industrial de Santander

23 de Septiembre de 2024

1. ¿Por qué el lenguaje de máquina es importante para definir la arquitectura computacional?

Antes de responder la pregunta, se realiza una definición de lenguaje de máquina y arquitectura computacional:

Lenguaje de máquina: Consiste en datos binarios (una serie de 0 y 1) que llegan de esta forma a circuitos microprogramables (en el contexto de computadores es leído por la CPU o procesador), los cuales interpretan esta secuencia de binarios como nivel alto de tensión o señal activa (1); o nivel bajo de tensión o señal apagada (0). El lenguaje de máquina reúne las instrucciones que recibe una máquina a la hora de llevar los procesos para los que haya sido programada. Estos permiten la comunicación usuario-máquina.

El lenguaje de máquina es específico de cada máquina o arquitectura de la máquina, puesto que este es un conjunto de instrucciones que determinan acciones que serán realizadas por la computadora en cuestión. En este sentido, un gran conocimiento sobre la arquitectura física del ordenador es requerido, por lo que este se vuelve imperativo para definir la arquitectura computacional. Las arquitecturas vistas en clase hasta ahora, como RISC, CISC, x86 o ARM, tienen su propio conjunto de instrucciones y características, por lo que el lenguaje

de máquina juega un papel crucial al definir el manejo de estas instrucciones de muy bajo nivel.

Con respecto a la arquitectura computacional, según la Universidad de Tiffin, Ohio (2023) [5], es un conjunto de reglas, principios y estándares que definen la estructura y el diseño de los componentes de hardware y software (el que se comunica directamente con el hardware) que conforman un sistema informático.

Partiendo de esta idea, en el libro *Computer Organization and Design* (2012) [1] se explica que los ordenadores poseen capas dentro de un modelo de abstracción, siendo el uso de estas una técnica principal al desarrollar sistemas de computadora sofisticados; dentro de este modelo, la abstracción de la interfaz entre el hardware y el software de más bajo nivel es de suma importancia. Dicha abstracción recibe el nombre de arquitectura del conjunto de instrucciones (Instruction set architecture) o ISA. Esta capa de abstracción posee lo que se necesita saber para que un programa binario en lenguaje de máquina funcione correctamente, incluyendo instrucciones, dispositivos I/O (como se pudo verificar en el proyecto 04 de Nand2Tetris) y demás cosas. Por lo tanto, estas instrucciones al determinar qué operaciones puede realizar el hardware, así como el modo en que se comunican los componentes del sistema, depende directamente de la programación en lenguaje de máquina que posea el procesador, por lo que define directamente la arquitectura del ordenador.

Asimismo, otras razones más triviales son la incidencia en la eficiencia del manejo del hardware, dado que en el lenguaje de máquina se diseña el formato de las instrucciones que maneja el procesador, así como el número de estas que es capaz de soportar y en los ciclos del reloj necesarios para que la CPU ejecute una tarea, por lo que es importante que el lenguaje tenga en cuenta la arquitectura del hardware que va a manejar (o, en su defecto, defina la misma); el modo de acceder a memoria y registros, entre otras. En síntesis, este establece las bases sobre las que se establecen la interacción entre el diseño de la ISA (interfaz hardware – software de bajo nivel) y la implementación física del hardware.

2. ¿Qué diferencia ven entre arquitectura computacional, arquitectura de software y arquitectura del sistema? Justifique su respuesta.

Antes de responder la pregunta, se parte de una breve definición de arquitectura de software y arquitectura de sistema (arquitectura computacional se encuentra definida en la primera pregunta).

Arquitectura de software: De acuerdo con los libros *Software Architecture in Practice* (2013) [6] y *Arquitectura de Software* (2016) [7], la arquitectura de software de un sistema es el conjunto de estructuras necesarias para razonar sobre el sistema, comprendiendo elementos de software, relaciones entre ellos y propiedades de ambos. Con elementos de software se hace referencia a la forma individual de las partes del sistema que se deben desarrollar (los módulos); mientras que sus propiedades son las interfaces: contratos que exhiben estos módulos y permiten establecer relaciones de dependencia entre ellos.

En este sentido, esta clase de arquitectura define la estructura y diseño, a alto nivel, de un sistema de software. Está relacionado con la estructuración de un sistema para satisfacer los requerimientos de los clientes y demás involucrados, con énfasis en los atributos de calidad. Tiene cierta similitud con la arquitectura computacional, puesto que la manera en que se estructura un sistema tiene impacto directo sobre la capacidad de este para satisfacer los requerimientos.

Arquitectura del sistema: Según la Universidad Internacional de La Rioja (s.f.) [8], la arquitectura del sistema describe el proceso más avanzado de diseño, desarrollo y estructura de los sistemas informáticos, incluidos subsistemas, dispositivos de red, hardware y software, describiendo conceptualmente los componentes y subsistemas físicos y lógicos conectados entre sí (incluye componentes de hardware y software). Por lo tanto, su objetivo es definir la estructura, vista y comportamiento de todo un sistema, a la par que colabora en definir una infraestructura que normalmente requiere un nivel de software no tan alto.

Considerando esto, la principal diferencia que hay entre los tres tipos de arquitectura ya descritos es su campo de acción: En el caso de la arquitectura computacional, se centra principalmente en cómo el software se comunica y maneja el hardware para cumplir con sus tareas, así como en el diseño de componentes de hardware; por otro lado, la arquitectura de software se enfoca en definir estructuras y el diseño internos del software para cumplir con requerimientos de calidad o comerciales, sin importarle cuál es el proceso mediante el cual el lenguaje base del software se comunica con el hardware (los procesos que ocurren a bajo nivel no hacen parte de su área de estudio); mientras que la arquitectura del sistema se

encarga de definir la infraestructura, así como los componentes de software que interactúan con esta, o sea, integra componentes de hardware y software.

Dicho esto, el nivel de abstracción requerido para cada tipo de arquitectura es distinto, dado que para la arquitectura computacional se requiere un nivel de abstracción de bajo nivel (o muy bajo), en la arquitectura de sistema se posee un nivel medio, incluyendo partes de no tan alto ni tan bajo nivel; por último, en la arquitectura de software se precisa de un nivel de abstracción alto.

Bonus: Como informático o computista: ¿La arquitectura computacional o la arquitectura del sistema no tiene en cuenta igualmente la arquitectura de software? Justifique su respuesta.

La arquitectura computacional no tiene en cuenta precisamente a la arquitectura de software, puesto que esta se centra principalmente en definir los modelos conceptuales y de abstracción del hardware. A pesar de que el lenguaje de máquina, el cual es el que maneja la arquitectura computacional, establece la relación o el puente entre el software y hardware, este no se encuentra dentro de los límites de la arquitectura computacional (aunque estén estrechamente relacionadas). Por otro lado, la arquitectura del sistema sí debe tener en cuenta la arquitectura de software que manejan los componentes presentes en su estructura. La arquitectura del sistema, al poseer un nivel de abstracción medio, tiene en cuenta tanto componentes de hardware como de software, por lo que entender la arquitectura del software es crucial para cumplir con su propósito de integrar de manera fiable y eficiente al hardware y software dentro de un sistema completo y funcional, por lo que posee la necesidad de optimizar la interacción del hardware con el software, especialmente en términos de sistemas operativos y controladores o drivers.

En conclusión, los niveles de abstracción de estos tres tipos de arquitectura son diferentes, pero eso no quiere decir que sean totalmente excluyentes. En el caso de la arquitectura de sistema, esta debe entender cómo interactúa los componentes del hardware y del software para poder cumplir con objetivos de rendimiento, seguridad y escalabilidad de la estructura de sistema que describe, así como de sus subsistemas.

Referencias

- [1] Patterson, D. & Hennessy, J (2012). *Computer Organization and Design. The Hardware / Software Interface. 4ta edición.* <http://wiki.sc3.uis.edu.co/images/3/34/TextBookFull.pdf>
- [2] Universidad Nacional Autónoma de México (s.f.). *Lenguajes de programación.* [https://repositorio-uapa.cuaieed.unam.mx/repositorio/moodle/pluginfile.php/2655/mod_resource/content/1/UAPA-Lenguajes-Programacion/index.html#:~:text=La%20funcin%20principal%20de%20los,mquina%20\(0%20y%201\)](https://repositorio-uapa.cuaieed.unam.mx/repositorio/moodle/pluginfile.php/2655/mod_resource/content/1/UAPA-Lenguajes-Programacion/index.html#:~:text=La%20funcin%20principal%20de%20los,mquina%20(0%20y%201))
- [3] Ceballos, F. J. (2004). *Enciclopedia del lenguaje C.* https://elhacker.info/manuales/Lenguajes%20de%20Programacion/C/kupdf.net_enciclopedia-del-lenguaje-c-ceballos-sierra-fco-javierau.pdf
- [4] Alonzo, J., Universidad de Guanajuato (2010). *Lenguaje de Programación.* https://www.cimat.mx/~pepe/cursos/lenguaje_2010/slides/slide_02.pdf
- [5] Sulbarán, I., Tiffin University (2023). *¿Qué es arquitectura de computadoras?.* <https://global.tiffin.edu/blog/que-es-arquitectura-de-computadoras>
- [6] Bass, L., Clements, P. and Kazman, R. (2013). *Software Architecture in Practice. 3rd Edition.* https://edisciplinas.usp.br/pluginfile.php/5922722/mod_resource/content/1/2013%20-%20Book%20-%20Bass%20%20Kazman-Software%20Architecture%20in%20Practice%20%281%29.pdf
- [7] Cervantes, H., Velasco, P., et al., CENCAGE Learning (2016). *Arquitectura de software – Conceptos y ciclo de desarrollo..* https://www.researchgate.net/profile/Perla-Velasco-Elizondo/publication/281137715_Arquitectura_de_Software_Conceptos_y_Ciclo_de_Desarrollo/links/57144e1408aeebe07c0641ab/Arquitectura-de-Software-Conceptos-y-Ciclo-de-Desarrollo.pdf
- [8] Universidad Internacional de La Rioja (s.f.). *¿En qué consisten las arquitecturas de software y sistemas?.* <https://colombia.unir.net/actualidad-unir/arquitectura-software-y-sistemas/#:~:text=Arquitectura%20de%20sistemas%3A%20Describe%20el,de%20red%2C%20hardware%20y%20software.>