

EntrevistasEnDiferido

Confirmado próximo entrevistado, se llama Isaac, es formador, autor de *El mundo de Bitman*, creador de www.architecnologia.es y mucho más. Comenzamos el próximo Lunes 19 de Abril. #entrevistasaac @andatux

Comenzamos una nueva entrevista a un formador, administrador y mucho mas, pero lo mejor es que se presente el mismo. ¿Te podrías presentar en unas líneas?

Hola a todos. Un placer. Gracias por la oportunidad.

Básicamente soy una mente inquieta que tiene una constante necesidad de aprender sobre tecnología. Concretamente sobre arquitectura de computadoras (y en especial sobre microprocesadores) y también sobre Linux, que al fin y al cabo es la capa que está estrechamente relacionada con la gestión de los recursos de hardware.

Desde que era niño, siempre me ha encantado desmontar juguetes para ver qué había dentro, y cómo funcionaban, además de intentar hacer multitud de inventos (algunos finalizaban enseñándote lecciones a la fuerza, con ciertos incidentes, como cuando descubrí que el grafito conduce mejor en un sentido, y presenta mayor resistencia en el otro sentido, por lo que hace que cuando pasas una corriente eléctrica se caliente...ya se puede imaginar cómo terminó aquello). Veo que esto de desmontar los juguetes y experimentar es algo común en muchos otros miembros del sector de la tecnología...

Esa necesidad de querer saber cómo funcionan las cosas es la que me ha llevado a investigar sobre arquitectura y sistemas operativos, los pilares del funcionamiento de las computadoras, como he comentado anteriormente.

Para empezar, unas cuantas preguntas cortas para conocerte mejor a ti y a tu entorno tecnológico.¿Qué ordenador utilizas?

Mi equipo... Generalmente me gusta montar mi propio PC eligiendo pieza a pieza. Pero por cuestiones económicas, hace un par de años necesité un nuevo equipo y encontré un AIO Lenovo con un precio muy bueno. No es nada del otro mundo, pero para el uso que le doy es suficiente. Básicamente es un AMD Ryzen 5 de 2ºGen, 8GB de RAM, y 256 GB SSD M.2 NVMe PCIe + 1TB HDD.

¿Que sistema operativo utilizas habitualmente?

En cuanto al S.O., desde hace años uso solo Linux. Empecé con SUSE, que venía en un CD de regalo en una entrevista de informática que me compró mi madre, y luego he ido pasando por diferentes distros hasta aterrizar en Ubuntu. Aunque estoy a disgusto con algunas cosas de GNOME que me están haciendo pensar en un cambio hacia KDE Plasma.

¿Lenguaje de programación?

Sobre el lenguaje de programación, de entre los pocos que sé, me quedo con C y VHDL. El primero por estar muy ligado al mundo Linux, y porque me parece más fácil que otros orientados a objetos (admiro a la gente que se le dan bien los OO, pero yo no soy uno de ellos). El segundo es para descripción de hardware, que como dije en la presentación es otra de mis pasiones.

¿Herramienta imprescindible en cualquier ordenador que utilizas?

¿Herramientas? En esto voy a ser algo aburrido, ya que por mi actual trabajo, básicamente me muevo con Firefox y LibreOffice. Últimamente no tengo tanto tiempo para otros software como algunos entornos EDA, compiladores,...

¿AMD o Intel?

Aunque mi primer microprocesador fue un Katmai, actualmente me quedo con AMD. Hay algunos ingenieros allí realmente buenos. De hecho, invirtiendo cantidades muy inferiores en I+D respecto a Intel, han podido hacer mucho "daño" a la competencia. Y eso te da una idea del talento de algunos arquitectos.

¿ARM o x86?

ARM vs x86... Aunque me quede con AMD, tengo que confesar que detesto profundamente x86 y sería muy positivo contar con alternativas en el mundo del PC. x86 tampoco es nada recomendable para aprender ensamblador por primera vez con ella, pese a que muchos la eligen para iniciarse en el ASM (claro error). ARM es una alternativa interesante desde muchos puntos de vista, por ejemplo por su relación FLOP/w, superficie del die (por tanto yield), etc. Y ojalá que RISC-V siga su tendencia y también se transforme en alternativa. Me atrevería decir que podría ser para el hardware lo que "Linux" ha sido para el software.

Una pregunta clásica en todas las entrevistas, que cambia en función del perfil del entrevistado. En tu caso, he escogido el ámbito de GNU/Linux. ¿Qué añadirías a GNU/Linux?

¿Añadir? Tal vez temas de seguridad. Por ejemplo, echo en falta una tienda de apps algo más cuidada, como las de los dispositivos móviles. A veces, en estas tiendas te puedes encontrar paquetes obsoletos o de proyectos ya abandonados. Y eso no es positivo de cara a la seguridad. Tampoco estaría de más un antimalware real, ya que los antivirus que existen suelen ser un port de las versiones de Windows, lo que significa que no tiene un motor para malware específico de Linux. Cuando hablas con algunos responsables de empresas desarrolladoras de antivirus y les preguntas por eso, suelen eludir el problema diciendo que sí detectan malware para Linux. Pero cuando tratas de preguntar a qué se refieren, ves que hablan de código multiplataforma como Java, etc.

¿Qué eliminarías de GNU/Linux?

¿Eliminar? La fragmentación. Es positivo que haya cierta variedad para satisfacer distintas necesidades, pero no tanta. A veces una simple disputa o desacuerdo entre desarrolladores

termina en un fork o un nuevo proyecto. Creo que una de las cosas que más urgen en GNU/Linux es la estandarización. Esa uniformidad acercaría a muchos desarrolladores. Por ejemplo, cuando le pregunté a Jeri Ellsworth sobre el soporte para Linux de sus gafas me dijo que ellos usaban Ubuntu internamente para el desarrollo, y era funcional. Sin embargo, no dieron soporte oficial para Linux. Me dijo que, si lo hacían, sería solo para Ubuntu, puesto que no querían empaquetar en varios tipos de binarios. Y eso es común a muchos desarrolladores. Los paquetes universales son un paso, pero se necesita más estandarización.

¿Qué modificarías de GNU/Linux?

¿Modificar? Pues... quizás la pila de red, que podría mejorar (véase la de BSD, que es bastante buena). La pila de audio también es mejorable (o modernizable mejor dicho). Las APIs también podrían modificarse para mejorar...

Aunque me encante Linux y tiene cosas extraordinarias (infinitamente superior a Windows), no se puede caer en el error de pensar que todo es perfecto. Hay que ser inconformista y mejorar aún muchas cosas. La lista sería bastante larga, pero poco a poco van apareciendo desarrollos muy interesantes que van reparando algunas de las debilidades...

Si tuvieras que escoger un procesador que su diseño ha implicado el mayor cambio en el mundo del hardware , de entre todos los desarrollado a lo largo de la historia. ¿Qué procesador escogerías?

Uff. No sabría decir uno... Por ejemplo, los DEC han sido una referencia en muchos sentidos (de hecho, muchos de sus responsables han terminado en AMD e Intel). El Alpha 21164 creó un interesante diseño de buffers en forma de árbol para distribuir la señal de reloj de forma más homogénea a lo largo del die, sin los problemas de delay.

Por otro lado está el MOS 6502, creado por ex miembros de Motorola que trabajan en el 6800. Y demostraron que se podían hacer microprocesadores baratos para la informática de consumo, y no solo para empresas. De hecho, resultó ser 7 veces más barato y 4 veces más rápido que el 6800... Eso propició que fuese el chip elegido por muchas máquinas de la época, como Atari, Apple, Commodore, etc.

El Intel Pentium Pro y AMD K5 también fueron muy interesantes por usar traducción de sus instrucciones CISC a unas estilo RISC llamadas micro-operaciones. Una tendencia que han seguido los modernos microprocesadores que usan instrucciones largas. Así pueden funcionar como un RISC en cuanto a su circuitería, con todas las ventajas que implica eso. De hecho, si no recuerdo mal, AMD usó como base del K5 su 29k, una de las familias de microprocesadores RISC más populares de la época. A ese 29k le agregarían un motor de traducción para convertir las x86 CISC.

Y... no sé, se me vienen a la cabeza también otros diseños interesantes. Algunos fueron unos adelantados a su época, y eso hizo que fracasaran. Como los INMOS Transputer (computación paralela), Transmeta Crusoe/Efficeon (VLIW y bajo consumo), Cyrix MediaGX (uno de los primeros intentos de SoC), etc.

Como has comentado, AMD tiene mucho talento y el salto de calidad que ha dado es enorme, siendo mucho más pequeña que Intel, le ha hecho mucho daño. Pero desde el punto de vista técnico. ¿Qué tiene Ryzen y su arquitectura para ser tan buenos?

Es complicado evaluar eso sin conocer detalles que las compañías no suelen dar. Pero parece que con Zen se han preocupado mucho por el Front-end (fetch, caché de instrucciones, predictor, decode...) de la CPU, algo que descuidaron en Bulldozer y microarquitecturas derivadas.

Anteriormente, muchos diseñadores han prestado bastante atención al Back-end, con más unidades de ejecución superescalares, agregando FPU's dedicadas a extensiones (SSE, AVX...), etc. Pero a esas unidades hay que alimentarlas para que puedan trabajar, y sin un buen Front-end que las mantenga ocupadas podría verse afectado el IPC.

Intel consiguió un buen rendimiento single-core así, y ahora AMD ha deshecho el error para volver al buen camino con un Front-end fuerte. Además, al dejar el modelo IDM y transformarse en una fabless, han podido apostar por el proceso ganador ahora, eligiendo la foundry TSMC. Por tanto, mientras Intel está atascado con su nodo de fabricación (por la obsesión de conseguir mayor densidad, experimentando con materiales "exóticos" para la capa M1 que producen problemas de adherencia, etc.), AMD ha podido usar la tecnología de TSMC de 7nm. Eso ha supuesto también un impulso (no hay más que fijarse, por ejemplo, en el TDP de Intel vs AMD).

Además, la competencia parece algo perdida. Intel (al menos en Rocket Lake) está intentando tapar estas carencias apostando por una curva f/v optimizada para no dar cifras alarmantes de TDP con FP y AVX-512, pasando por alto el principio "*make the common case fast*". Ha hecho crecer el Back-end, con unidades de coma flotante grandes, lo que ha hecho que crezca la superficie del die. Quizás por eso solo puedan integrar hasta 8 núcleos, mientras en su predecesor (Comet Lake) podían llegar a los 10 cores para el chip monolítico sin afectar tanto al yield... Mal camino, recuerda un poco a la era AMD Fusion. Pero es lo que pasa cuando los inversores mandan más que los ingenieros en una empresa...

En definitiva, los nuevos diseños, para ser competitivos, necesitan mimar el Front-end y un buen nodo de fabricación. Véase por ejemplo el Apple M1. Una microarquitectura bastante poco sorprendente (y con graves debilidades, como un deco muy largo que obliga al compilador a hacer malabares), pero con una maravillosa caché que reduce la tasa de fallos, haciendo trabajar "poquito" al TLB (por tanto, sin tantos ciclos de penalización al traer instrucciones).

Has comentado anteriormente sobre RISC-V, como una alternativa a ARM, pero... ¿Qué es RISC-V y porque es tan importante que triunfe?

RISC-V no es una CPU de código abierto, como algunos creen. Es una ISA, como lo es MIPS, POWER, SPARC, IA-64, x86, ARM, etc.

Por tanto, se podría implementar mediante diversas microarquitecturas de CPU. Y estos núcleos pueden ser propietarios o de código abierto. Voy a tratar de explicarlo con unos ejemplos, para que sea algo más intuitivo...(o eso espero).

Por ejemplo, un Athlon es una CPU con microarquitectura K7 propiedad intelectual de AMD. Una microarquitectura no es más que una implementación de la ISA x86. Esta ISA es propiedad de Intel en este caso, y que AMD puede usar por una serie de acuerdos de cruces de patentes, etc. Intel, de esta forma, puede vetar quién usa, y quién no, esta ISA. Además, el núcleo del Athlon también es IP, propiedad intelectual de AMD. Si otros quisiesen usarlo, AMD debería de licenciarlo y le deberían pagar las regalías.

Luego está el caso de Arm (ahora propiedad NVIDIA), que licencia tanto la ISA como también núcleos IP. Por ejemplo, un Mediatek Helio X20 usa núcleos IP licenciados por Arm. Por tanto, Mediatek no tiene que diseñarlos desde cero, simplemente paga y obtiene el diseño para integrar los núcleos Cortex-A72 y A53 en su SoC. Apple M1 es un caso totalmente diferente. En el caso de Apple han pagado por la licencia de la ISA ARM para poderla usar, pero no han comprado núcleos IP. Han sido ellos mismos quien han diseñado desde cero la implementación de la ISA (microarquitectura).

En el caso de RISC-V, al ser open-source (licencia BSD), no estaría controlada por empresas como NVIDIA o Intel. Cualquiera puede hacer una implementación, sin restricciones (tampoco por guerras geopolíticas) o regalías.

Eso es muy positivo para paliar la dependencia tecnológica de Europa, que tras haber perdido ARM, no tendríamos ninguna ISA importante sobre la que desarrollar procesadores competitivos. Ya hay un proyecto de un acelerador para supercomputación (véase EPI/SiPearl, hay información en mi blog), que se prueba en el Marenostrum de Barcelona. Y también ha comenzado el proyecto eProcessor, para un núcleo de CPU 100% europeo también para supercomputación. También se ven otros proyectos interesantes como GAIA-X, para deshacerse de las GAFAM/BATX. Europa marcha así hacia la no dependencia de China y EE.UU., lo cual es muy positivo.

Eso en cuanto a la licencia. Pero además de eso, le veo otros puntos fuertes:

- Es moderna, pensada también para la IA y otros paradigmas actuales. Cosa que otras ISAs no lo están, ya que muchas fueron desarrolladas hace décadas.
- Es modular, por lo que se pueden agregar los módulos en función de lo que necesites (muy positivo, por ejemplo, para crear aceleradores para HPC). Incluso permite agregar un subconjunto de instrucciones personalizadas. Es decir, lo contrario a la x86, que es incremental, lo que hace que sea cada vez más extensa y compleja para mantener la retrocompatibilidad de software... Si no recuerdo mal, la ISA x86 de 1978 (del 8086) tenía 80 instrucciones, actualmente están próximos al millar y medio (algunas inútiles, puesto que no se usan actualmente). Eso significa complejidad en la unidad de control > menor agilidad > mayor superficie > más costes etc.
- Es segura, ya que se ha creado teniendo en cuenta los desafíos actuales, con un sistema de 3 anillos para el aislamiento (Matizo: esto no es garantía de que la microarquitectura resultante sea segura). Y ofrece mayor transparencia con respecto a una ISA propietaria (p.e.: recientemente se han descubierto 2 instrucciones ocultas y no documentadas en CPUs Intel).

Por cierto, he hablado de los núcleos IP anteriormente. Y, aunque en el pasado no lo hicieron, ahora sí que lo han hecho. AMD licenció núcleos IP Zen 1 a China. El resultado fueron las CPUs Hygon Dhyana, que es una variante del EPYC destinadas a HPC.

También existen diseños custom de AMD. Por ejemplo los chips de las videoconsolas Xbox y PlayStation. Pero es algo diferente, no han sido licenciados, sino que se han personalizado para estas consolas en un trabajo conjunto entre Microsoft-AMD y Sony-AMD respectivamente.

Intel también parece que ha anunciado recientemente que licenciará sus núcleos IP. Será interesante ver quién los va a usar...

Y me gustaría matizar otra cosa. Además de AMD, la x86 también la usan otras empresas, que deben pagar por ello. En el pasado lo hicieron IDT, Cyrix, etc. Y ahora VIA Technologies sigue haciéndolo. De hecho, tienen un join venture con el gobierno de Shanghai. El resultado es una fabless llamada Zhaoxin que está diseñando microprocesadores multinúcleo x86-64 para el mercado chino. Y ojo con ellos, porque algunos ya casi igualan o superan a Intel Core i7 de generaciones atrás (6°/7°Gen).

Y ya paro, porque el tema da para muchísimo más... Podría comentar muchas cosas. Como los modelos de licenciamiento de SPARC. O la apertura de ISAs como MIPS y POWER,... Tal vez curiosidades como que el IBM PowerPC 615 podía usar de forma nativa instrucciones de la ISA PPC y x86 (tenía un núcleo x86). O que Rusia también hizo algo similar en sus chips E2k, pero en este caso usando traducción dinámica para x86. O cosas como que hubo clones no autorizados de chips de Intel (especialmente en el bloque soviético). Si queréis preguntar algo concreto, dejad comentarios e intento responder.

En tu blog veo hablas de formación y certificaciones, siempre he tenido dudas donde encajar las certificaciones dentro del aprendizaje de GNU/Linux. ¿Dónde realmente son útiles? ¿Al comienzo de un aprendizaje(partir de cero conocimientos), para reciclarse de otro ámbito, para aumentar tus conocimiento en una materia, para todos los casos mencionados o para otro caso que no he mencionado?

Yo creo que pueden ser positivas en cualquier momento. Más aún en un mundo tan cambiante como el de la tecnología. Siempre debes estar actualizado.

En cuanto a partir de cero, en principio no habría problema en algunos cursos. Pero, por experiencia, me he dado cuenta que algunos alumnos que he tenido, y que partían de un nivel bajo, les cuesta mucho y no llegan a sacarle todo el partido al curso, en comparación con los alumnos que ya tenían una buena base. Además, algunos se esfuerzan mucho y hacen por aprender, pero otros te contestan la primera burrada que se les pasa por la cabeza en los exámenes. Estos últimos suelen ser los que simplemente quieren hacer el curso para tener algo más que agregar al CV, y no les importa realmente el conocimiento.

En cuanto a las certificaciones LPI (especialmente LPIC-2 y las tres variantes de LPIC-3), o las de la Linux Foundation (además, tienen otras adicionales para los interesados devops,

full-stack, Kubernetes,...), están bastante bien valoradas en el sector. Te pueden servir para reciclarte o para partir de cero (si te esfuerzas). Pueden ser muy positivas para los futuros administradores de sistemas, etc.

El problema es que en las ofertas de empleo suelen tener 2 cosas llamativas:

- 1 Pedir siempre experiencia. Pero me pregunto quién puede tener experiencia si no dan oportunidades a los que no la tienen.
- 2 Y los que no la piden, suelen tratar de aprovecharse en algunos casos. Piden un full-stack con sueldo bajo. O un sysadmin Senior pagándote como un Junior.

Y en algunas ofertas ocurren ambas... Realmente te topas con cada cosa que te dejan impresionado.

Yo por ejemplo, hace años (2006), me topé con un directivo de una inmobiliaria de Cádiz (no diré el nombre) que iban a montar su propio servidor. Y necesitaban un administrador. Contactó conmigo desde el formulario de mi web, para que se lo hiciera yo en remoto. Quería que estuvieras dedicado desde las 8:00 de la mañana (incluidos sábados y domingos), no solo al servidor, sino a todos los problemas que tuviese (y su nivel era tal, que me llegó a preguntar cómo se podía guardar una carpeta en un pendrive). Es decir, cada dos por tres estaba preguntando cosas... Evidentemente, todo eso por un sueldo vergonzoso. Ese contacto se produjo un viernes, el lunes lo rechacé. Pero algo no debió entender muy bien, ya que todos los días me mensajeaba para preguntar dudas. Quizás debió pensar que si por el sueldo mísero no acepté, tal vez estaría más feliz siendo su helpdesk gratis...

Y finalizo con un consejo. A veces no es solo mirar ofertas de trabajo. A veces también hay que llamar a las puertas. El 99% quizás no se abran, pero algunas lo harán. No hay que desistir.

Además, aconsejaría a todos que miren alto. Veo mucha gente que tiene sueños u objetivos altos y, una vez terminan sus estudios o su carrera universitaria, y terminan conformarse con trabajar en la tienda de informática de su barrio, o en cualquier empresa de la zona. Aparcan sus sueños y se conforman. Y eso no debería de ser así.

Por ejemplo, tal vez sueñas con trabajar algún día en grandes centros de datos, como los de Google, AWS, etc. Si comienzas a luchar y llamar a la puerta de las empresas más relevantes del sector, tal vez no se abrirá la puerta de la inmensa mayoría (o tal vez sí), pero algunas sí se abrirán. Incluso si tu puesto dentro de esa empresa no es el que esperabas, piensa que una vez dentro, cuando te abren la puerta y has metido el pie para que no se cierre, puedes escalar. En el peor casos tal vez te veas un día trabajando para OVHcloud. No es lo que buscabas en un principio, pero seguro que es mejor que la tienda de tu barrio. ¿No crees?

En el mundo de la supercomputación ARM va entrando con paso firme , cada vez hay mas supercomputadores del top500 que llevan ARM de forma exclusiva o junto x86 (Xeon y Epyc). ¿Cuál es el futuro de la supercomputación, solo ARM, arquitectura hibrida ARM y x86, una convivencia "pacifica" u otra opción?

Alpha, PA-RISC, IA-64, SPARC, MIPS, etc. han ido poco a poco desapareciendo del sector y dejando paso a un dominio x86 casi absoluto, y actualmente es un ámbito casi exclusivo de Intel y AMD. IBM POWER sigue manteniendo una pequeña cuota, y ahora entra ARM en el juego.

El supercomputador nipón Fugaku, con los chips A64FX, han demostrado que se puede llegar al puesto 1 del Top500 con un procesador basado en Arm. Y poco a poco van entrando más procesadores de esta familia. Entre otras cosas, porque están usando el ecosistema x86. Es decir, los ARM se han adaptado para poder funcionar con GPUs hechas en principio para x86, como las de AMD y NVIDIA, además de ser compatibles con UEFI, ACPI, y otras tecnologías y estándares x86. Es decir, para poder luchar contra x86, se están nutriendo también de su propio ecosistema, por la enorme complejidad y tarea tan árdua de construir uno desde cero (RISC-V está haciendo lo mismo).

No van a reinar a corto plazo. x86 seguirá dominando, entre otras cosas por todo ese software que habría que portar (aunque ya existen muchos paquetes)...

Pero lo positivo de Arm es su eficiencia energética FLOPS/w (muy importante en este sector) y su reducido tamaño. Ese reducido tamaño del núcleo permite usar más núcleos en comparación con otros núcleos de gran tamaño. Por tanto, es otra gran ventaja. Se pueden hacer chips con decenas de núcleos (48, 72,...), lo que puede conseguir rendimientos muy buenos, reducción de costes de producción, mejora del yield, etc.

Dicho eso, supongo que x86 irá cediendo poco a poco ante ARM y RISC-V. Pero también es cierto que lo que cada vez tiene también más peso es el uso de GPGPUs y la computación heterogénea.

Además, hay que prestar especial atención a los DSA. Van a tener cada vez más relevancia. Las CPUs convencionales irán acompañadas de aceleradores específicos en un FPGA o ASIC. Apple ya lo usa en sus A-Series, y M1 (véase Neural Engine).

Y por otro lado está la computación cuántica. Otro paradigma que también ayudará mucho a la supercomputación.

Por tanto, creo que no será tan importante ARM o x86, sino buscar las técnicas más eficientes energéticamente para tareas específicas.

En tu página web tiene un apartado de libros, donde hay una publicación muy interesante que me gustaría preguntarte por ella. ¿Qué es *El Mundo de Bitman*?

De eso tuvo un poco la “culpa” mi madre. Recuerdo que un día me regaló una revista con un reportaje especial sobre microprocesadores. Tras leerlo me “enamoré” de ese componente. Después de eso, comencé a investigar con los pocos medios que en aquella época tenía. Incluso aprovechaba algunos tiempos en los que, el profesor de la asignatura de informática del instituto, nos dejaba para hacer lo que quisiésemos. Otros alumnos consultaban sus correos, chateaban, miraban resultados de fútbol, etc. Yo me dedicaba a

descargar información en un disquete para leerla en casa tranquilamente (en casa no tenía Internet en esa época).

Desde entonces, comenzaría a aprender e investigar sobre el tema. Inicé una especie de apuntes para mi. Poco a poco iban creciendo y mejorando. Compré multitud de libros en español e inglés, descargué datasheets, documentos de trabajos universidades, y un sin fin de fuentes más.

Tras 15 años investigando, pude finalizar esos apuntes y una base de datos con miles de microprocesadores, desde el primero (CADC), hasta los más recientes, pasando por algunos extraños y poco conocidos. En principio, era algo para mi, como he dicho. Pero me animaron a publicarlo.

El resultado fue una enciclopedia sobre microprocesadores de unas 3500 páginas, 8 volúmenes y unos 9 kg de peso. Eso es *El Mundo de Bitman*.

La base de datos no la he publicado aún, pero últimamente estoy meditando liberarla para que otros puedan usarla y colaborar en su ampliación (tengo que pensar sobre la licencia y formato). Tiene algo más de 1400 familias de microprocesadores (no incluyo los modelos derivados), y columnas con su generación, fecha, diseñador, fabricante, frecuencia, multiplicador/divisor, tamaño de palabra, número de transistores, tecnología de fabricación y materiales, familia lógica, voltaje, tipo de ISA, ISA, extensiones, si integra FPU y GPU, tipo de empaquetado, socket/número de pines, tamaños de caché, núcleos, tecnologías de gestión de energía, seguridad, MIPS, Stepping, dimensiones del die, y un último campo de descripción con las curiosidades.

Como última pregunta de la entrevista , el fin de semana cambia el formato , me gustaría saber tu opinión sobre el futuro que nos espera. ¿Cuál crees que es el futuro en los microprocesadores?

Soy muy malo haciendo predicciones. De hecho, en mi enciclopedia acerté con el futuro de los ARM, pero supongo que es la excepción que confirma la regla. Por eso esto es simplemente una opinión personal:

- 1 En cuanto al tema físico, pensaba que se llegaría pronto al límite de la tecnología del silicio. En cambio, van apareciendo nuevas tecnologías (estructuras y materiales) que están extendiendo más y más los chips basados en silicio. Tras los MOSFET con tecnología CMOS planar han llegado las estructuras FinFET, y después GAA, etc. Algo similar al o que ocurre con las ciudades, que han comenzado a construir rascacielos para ocupar el mínimo de superficie. También se están haciendo cosas desde el lado del package, con los nuevos empaquetados 3D (véase InFO, LSI, CoWoS, EMIB,...) que permiten amontonar un die sobre otro, y también acercar la memoria al procesador para reducir la latencia (véase lo que ya se hace en algunos chips con la HBM). Me gustaría agregar la tecnología óptica, de forma similar a la fibra óptica que ha sustituido al cable de cobre en las redes, pero a nivel nanoelectrónico. Ya se está experimentando con ella, para enlazar mediante luz de forma mucho más rápida. Tanto entre unidades de procesamiento, como entre éstas

y la memoria. Tengo mis dudas respecto al 3D Stacking, puesto que eso implicará amontonar mucha densidad de potencia que hay que disipar. Supongo que va a ser todo un reto para los ingenieros mantener las pilas a temperaturas seguras. Pero viendo algunas presentaciones de Jim Keller parece que seguirá la tendencia por bastante tiempo, así que el silicio aún no va a llegar a su techo. De hecho, François Piednöel me dijo que él no viviría para ver el fin de esta tecnología, y que piensa vivir muchos años...

- 2 Desde el lado de la ISA y la microarquitectura, cada vez es más complicado dar grandes saltos de rendimiento, pero creo que se tenderá a diseños manycore y se buscará "ayuda" externa con DSAs para acelerar tareas específicas (y reducir la cantidad de energía consumida en movimientos de datos). Las próximas CPUs irán acompañadas de un FPGA con este tipo de aceleradores. También la IA puede contribuir más allá de las unidades de predicción de los microprocesadores. Hay empresas que tienen proyectos muy interesantes, como Cerebras, Tenstorrent, Graphcore, Mythic, etc.
- 3 La optimización se volverá fundamental también en el lado del software.

De lo que estoy bastante seguro es de que no habrá procesadores cuánticos a corto/medio plazo en los hogares como algunos piensan. Aunque sí se podrán usar como un servicio más en la nube (QaaS). Es más, espero equivocarme (puesto que esto reduciría el control del usuario sobre lo que se hace), pero por algunas conversaciones que he mantenido con gente de la industria de empresas muy relevantes (IBM, Red Hat, SuSE,...), creo que en un futuro todos tendremos "dumb terminals" (terminales tontos), simples dispositivos clientes, y todos los procesos se ejecutarán en la nube... Es decir, nuestros datos estarán en la nube, los programas se ejecutarán en la nube, etc. En nuestros dispositivos del futuro tan solo podremos ver los resultados. ¡Ojalá me equivoque! Y si no es así, espero que esto tarde bastante en llegar, puesto que sería el fin del poco control de los datos, privacidad y anonimato que ahora tenemos...

El fin de semana ,cambia el formato de la entrevista y el formato de las preguntas, con dos preguntas algo diferentes. La primera pregunta sería: ¿Qué te hubiera gustado que te preguntase? Evidentemente , debes responder a tu propia pregunta. La segunda pregunta. ¿Qué le hubieras preguntado a un formador?

La verdad es que no sé... Siempre hay más cosas que se quedan atrás. Pero es complicado decantarse por una. Me gusta mucho comentar sobre el tema de la fabricación de chips. Me apasiona. Así que tal vez me hubiera preguntado cualquier cosa relacionada con eso. Por ejemplo, algo que suscita interés: ¿Qué es el binning?

Si alguien quiere preguntar cualquier cosa sobre la fabricación de un chip, puede dejar su pregunta... Y dicho eso, paso a contestar mi pregunta:

De forma muy muy resumida, el binning es lo que hace que chips idénticos fabricados en una misma oblea se etiqueten con SKU diferente.

Por ejemplo, una oblea se hace crear cientos de chips AMD Ryzen 7 3800. Pero por defectos o variaciones durante los procesos de fabricación, no todos pueden funcionar de

forma estable en una frecuencia a un voltaje determinado, ni tampoco tienen todos sus núcleos funcionales.

Por tanto, tras unos tests se etiquetan diferentemente. Por ejemplo, solo los que pueden llegar a 3.9Ghz nominales (estables a un determinado voltaje) serán marcados como 3800X, si no llegan se bajará la frecuencia y se etiquetan como 3700, y si tienen 2 núcleos disfuncionales se desactivan y se etiquetan como Ryzen 5, con 4 núcleos funcionales Ryzen 3... Y así.

Esto no solo ocurre con AMD, también con Intel. E incluso en chips GPU, de memoria, etc. Una forma de aprovechar cuantos más chips mejor.

Sobre la pregunta que le haría a un formador, yo lo diría lo que creo que es más importante: ¿Cómo motivas a tus alumnos?

A un futuro entrevistado con ese perfil le haré esta pregunta que propones.

El Domingo, último día de la entrevista, no hay preguntas, será el momento de la despedida y que me digas tus métodos de contacto y cualquier otro proyecto que quieras promocionar, tienes este espacio para hacerlo.

Muchas gracias por la oportunidad. Ha sido divertido contestar las preguntas y poder compartir estos temas que no suelen tratarse con tanta frecuencia como otros temas relacionados con la computación.

Todos los interesados en seguirme o consultar dudas, me tenéis en:

Telegram @andatux

MyPublicInbox: <https://mypublicinbox.com/iandatux>

Twitter: @_iandatux_

Por último me gustaría que me recomendaras a alguna persona que creas que estaría dispuesto a participar, es muy importante que tenga cuenta en Telegram.

Propongo a Nuria (@ventress) para la próxima entrevista. Una maker, y excelente profesional de la seguridad. Existen muchos expertos en la materia bastante "mediáticos", cuyas opiniones a veces no están a la altura de su fama, pero tenéis que conocerla a ella. He observado muchas de sus opiniones y es certera como un arma de precisión. Da en el blanco, mientras otros parecen algo perdidos. Seguro que tiene muchísimas cosas interesantes que compartir...

¡Hasta la próxima!