"Un lenguaje de programación es de bajo nivel cuando sus programas requieren atención a lo irrelevante' Lenguaje de bajo nivel Bajo nivel: "Más cerca del metal" Bajo nivel vs alto nivel Alto nivel: Más cerca del pensamiento humano Las vulnerabilidades Spectre v Meltdown nacen del intento de construir procesadores rápidos y que tuvieran la misma máquina abstracta que un PDP-11, en la busqueda de un ILP más alto fue la causa directa. Código C proporciona una máquina abstracta principalmente en serie. Los procesadores que quieren mantener sus unidades de ejecución ocupadas confían en el paralelismo a nivel de **Procesadores** y GPU en el ejecución (ILP), esto agrega comlejidad para permitir una programación secuencial. paralelismo Las GPU tienen rendimiento alto sin uso de estas lógicas a cambio de necesitar programas explícitamente paralelos. **Emuladores** Procesador Intel moderno puede tener hasta 180 Procesador rápidos de instrucciones en vuelo a la vez. Intel vs PDP-11 máquina Máquina abstracta C secuencial tiene que abstracta C esperar que cada operación se complete antes secuencual de iniciar con la siguiente. El caché está oculto para el programador y su uso ediciente es una forma de hacer que un código se ejecute rápidamente en un procesador moderno. Caché En la máquina abstracta esto está completemante oculto, por lo que los programadores deben conocer laimplementación del cahé para hacer un código eficiente. Una característica común de los lenguajes de bajo nivel es que son rápidos sin necesidad de un compilador complejo para traducir a código rápido. Compiladores En C es necesario un compilador complejo para tener un mejor rendimiento, por ejemplo el compilador de Clang, el cual tiene aproximadamente 2 millones de líneas de código. Para procesar grandes cantidades de datos se necesita de un ciclo para hacer el procesamiento de cada elemento, para poder optimizarlo se necesita determinar que las iteraciones son independientes, que las lecturas de uno no interfieran en otro. **Bucles** Una vez independizados se necesita vectorizar el resultado, ya que se obtiene mayor rendimiento en código vectorial que en escalar (los lenguajes de bajo nivel tienen tipos de vectores nativos de longitudes arbitrarias. Optimización C garantiza que estructuras con el mismo preefijo se pueden usar indistintamente y expone el desplazamiento de de los campos de estructura en el lenguaje, con esto el compilador no de C puede reordenar campos para mejorar la vectorización. **Estructuras** En un lenguaje de bajo nivel el control detallado sobre el diseño de la estructura es una característica. SROA (Reemplazo Escalar de agregados), intenta reemplazar structs con variables individuales, esto permite que se trate a los accesos como operaciones independientes. Optimizaciones principales Anulación de bucle, ttransforma un bucle que contiene un condicional en un condicional con un bucle en ambas rutas, esto cambia el control de flujo, aunque puede causar comportamiento indefinido. Solo con un compilador lo suficientemente inteligente y rompiendo algunas reglas del lenguaje se puede hacer que el código en C se ejecute rápidamente. Un atributo clave de un lenguaje de bajo nivel es que los programadores facilmente entienden como la maquina de lenguaje abtracto mapea a la maquina fisica subyacente Algunas implementaciones de C han tenido que volverse más complejas para hacer ilusión que C se asigna facilmente al harware subyacente y proporciona código más rápido. Perminte la implementacion para insertar relleno en estructuras (no en arreglos) para asegurar que todos los archivos tengan un valor El uso de punteros vuelve complicado el enetendimiento de C Entendiendo C Cada palabra es algún dato o la dirección de algún dato. Vulnerabilidades de seguridad como el desbordamiento de enteros con signo (comportamiento indefinido de C) Dificil entendimiento de como un programa C se mapea en la arquitectura adyacente para el programador. La memoria es una matriz plana de celdas de almacenamiento indexadas por dirección. Aún cuando en la comparación bit a bit El compilador era libre de determinar que dos punteros con diferentes resultados malloc siempre se comparan como no iguales diga que describe la misma dirección. Dejar de intentar que C sea más rápido y prestar atención a los modelos diseñados para ser mas rápidos. Los programas en C tienden a tener pocos subprocesos ocupados SVE (Scalar Vector Extensions) brinda un vistazo a Espera a que el programador describa el grado de paralelismo una mejor interfaz entre el programa y el harware disponible y se basa en el harware para asignarlo. Imaginando un Procesador

que no sea C

C no es un lenguaje

de bajo nivel

Project Maxwell propuso un recolecto de basura de generacion joven que se ejecutaria en el cache y permite que la memoria se reciclara rápidamente.

Debido a la gran cantidad de código C heredado, los sistemas diseñados puramente para la velocidad, ejecutar un códgo C en estos seria problemático.

C no mapea el harware moderno muy bien.