

Curso Intensivo en Herramientas de Simulación de Procesadores (GEM5)

Esteban Mocskos (Profesor Depto de Computación) con la colaboración de Elba Garza (Assistant Teaching Professor, Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA USA), Tamara Silbergleit Lehman (Assistant Professor, Electrical, Computer and Energy Engineering, University of Colorado Boulder, USA) y Jason Lowe-Power (Associate Professor, Department of Computer Science, University of California, USA)

Programa:

Adquirir conocimiento de la herramienta GEM5, sus componentes y funcionalidades con el fin de poder utilizarla de manera efectiva en la investigación y desarrollo de sistemas computacionales.

Desarrollar la capacidad de modelar y simular diferentes componentes de una plataforma de cómputo, desde la arquitectura del procesador hasta la memoria.

Aprender a analizar los resultados obtenidos de las simulaciones para evaluar el desempeño de los sistemas ante distintos escenarios y condiciones.

Comprender cómo utilizar GEM5 como una herramienta fundamental para la investigación en arquitectura de computadores, permitiendo la exploración de nuevas ideas y el diseño de sistemas más eficientes.

Temario:

- Introduction
 - Introducción a la Arquitectura del computador
 - Introducción a la investigación en arquitectura del computador
 - Ejemplo de una idea de investigación: *secure memory* (memoria segura)
 - ¿Por qué necesitamos simuladores?
 - Conceptos iniciales en simulación
 - Qué es hacer simulación y por qué es importante.
 - Historia de GEM5
- Comenzando con el simulador GEM5
 - Ambiente *codespace*
 - Ejecutando la primera simulación
 - Ejercicio práctico: completar el código y ejecutar una simulación
- Usando GEM5
 - La biblioteca estándar (*standard library*) de GEM5.
 - Vista rápida de las ideas por detrás de la *stdlib* (*board, processor, cache hierarchy, memory*)
 - Ejercicio Práctico: crear una simulación simple con un procesador ARM.
 - Salida del simulador: Ejecutar un trabajo real en *SE mode* y analizar las estadísticas
 - Presentación de los componentes disponibles en la *stdlib*.
- La idea por detrás del simulador
 - Funciones útiles (*set_max_ticks/instructions*).
 - Modelando memoria con GEM5.
 - Modelos de memoria disponibles en GEM5.
 - Generadores de tráfico
 - Ejercicio práctico: usar el generador de tráfico para probar la memoria.
 - Ejercicio práctico: crear un nuevo generador de tráfico propio.

- Memoria cache
 - Modelando memoria cache en GEM5.
 - Modelos de memoria cache en GEM5 (*ruby* y *classic*).
 - Ejercicio práctico: una memoria cache clásica de tres niveles.
 - Políticas de reemplazo.
 - Políticas de marca (*tagging*).
 - Comparación entre modelo *classic* y *Ruby*.
 - Ejercicio: Ejemplo de uso de una jerarquía tipo *Ruby*.
 - Análisis de las estadísticas generadas en una ejecución.
- Modelando núcleos (*cores*)
 - Tipos de modelos de procesadores en GEM5.
 - Ejercicio práctico: comparación entre procesadores tipo *timing* y atómico. Análisis de las estadísticas generadas en cada caso.
 - Ejercicio práctico: crear dos núcleos propios tipo fuera de orden (*out-of-order*) y compararlos.
 - Predictores de saltos.
- Presentación de las diferentes *ISAs* y sus características
 - Uso de los recursos disponibles en GEM5: *workloads* y *suites multisim*.
 - Ejercicio Práctico: ejecución de una suite y verificación de los diferentes resultados de acuerdo a las aplicaciones utilizadas.
 - Ejecutando aplicaciones completas en GEM5.
 - Introducción al modo de emulación de *syscall*.
 - La utilidad *gem5-bridge* y el proceso de *cross compiling*.
 - Ejercicio Práctico: crear su propio *workload*.
- Emulación de sistemas completos (*Full system*).
 - ¿Qué es una emulación *full system*?
 - Conceptos básicos para arrancar un sistema real usando GEM5.
 - Ejecución en modo FS.
 - *m5term* para interactuar con un sistema en ejecución.
 - Comando para actuar ante la pregunta ¿Qué hacer cuando Linux arranca?
 - Estableciendo un workload para los dispositivos de almacenamiento
 - Presentación de eventos de salida y los tipos posibles.
 - Ciclo de simulación.
 - Crear imágenes en disco usando *packer* y *qemu*.
 - Extender o modificar una imagen en disco de GEM5.
 - Técnicas para acelerar la simulación
 - Procesadores *switchable*.
 - KVM *fast forwarding*.
 - Ejercicio práctico: ejecutar una simulación usando KVM, *switching*, eventos de salida complejos y medición de tiempos.
 - *Checkpointing*.
 - Ejercicio Práctico: ejecutar a partir de un checkpoint usando diferentes configuraciones.
- Simulaciones basadas en *sampling* con GEM5.
 - Simpoint ideas
 - Simpoint analysis
 - Simpoint checkpoints
 - ¿Cómo analizar resultados de simulaciones basadas en *sampling*?
 - Ejercicio Práctico: ejecutar una simulación basada en *simpoint*.
 - Loopoint/Elfies

- Ejercicio práctico: Análisis estadístico y ejecución de simulaciones.
- Modelos de consumo
- Ejercicio práctico: ejecutar una simulación enfocada en evaluación de consumo.
- Desarrollo de modelos en GEM5
 - Introducción a SimObject
 - Ambiente de desarrollo, estilo de codificación.
 - El SimObject más simple. Ejecución, debugging.
 - Simulación basada en eventos
 - Creación de un evento tipo *callback* simple.
 - Eventos de *scheduling*.
 - Modelar eventos de ancho de banda y latencia.
 - Modelando núcleos.
 - Nuevas instrucciones.
 - Debugging
 - Modelar coherencia de cache con Ruby y SLICC.
 - Detalles de Ruby y estructura de SLICC.
 - Ruby network.
 - Extendiendo GEM5.
 - Puertos y *memory-based* SimObjects.
 - Conceptos de puertos (*request/response*) e interface de paquetes
 - CHI protocol

Bibliografía:

- Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K. Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R. Hower, Tushar Krishna, Somayeh Sardashti, Rathijit Sen, Korey Sewell, Muhammad Shoaib, Nilay Vaish, Mark D. Hill, and David A. Wood. 2011. The gem5 simulator. SIGARCH Comput. Archit. News 39, 2 (May 2011), 1–7. <https://doi.org/10.1145/2024716.2024718>
- John L. Hennessy, David A. Patterson, Christos Kozyrakis. Computer Architecture: A Quantitative Approach by J. L. Hennessy and D. A. Patterson. 7th Edition. Morgan Kaufmann; 7th edition (May 15, 2025)
- Sarah Harris, David Harris. Digital Design and Computer Architecture, RISC-V Edition. Morgan Kaufmann; 1st edition (October 22, 2021).
- Stuart Sutherland. RTL Modeling with SystemVerilog for Simulation and Synthesis: Using SystemVerilog for ASIC and FPGA Design. CreateSpace Independent Publishing Platform (June 10, 2017).
- David A. Patterson and John L. Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2nd Edition. Morgan Kaufmann; 2nd edition (December 31, 2020)
- Gueron, S. Memory Encryption for General-Purpose Processors. IEEE Security & Privacy, 14, 54-62. <https://doi.org/10.1109/MSP.2016.124>. (2016).
- Henson, M., & Taylor, S. Memory encryption. ACM Computing Surveys (CSUR), 46, 1 - 26. <https://doi.org/10.1145/2566673>. (2014).
- Lowe-Power, J., Ahmad, A., Akram, A., Alian, M., Amslinger, R., Andreozzi, M., Armejach, A., Asmussen, N., Bharadwaj, S., Black, G., Bloom, G., Bruce, B., Carvalho, D., Castrillón, J., Chen, L., Derumigny, N., Diestelhorst, S., Elsasser, W., Fariborz, M., Farmahini-Farahani, A., Fotouhi, P., Gambord, R., Gandhi, J., Gope, D., Grass, T., Hanindhito, B., Hansson, A., Haria, S., Harris, A., Hayes, T., Herrera, A., Horsnell, M., Jafri, S., Jagtap, R., Jang, H., Jeyapaul, R., Jones, T., Jung, M., Kannoth,

S., Khaleghzadeh, H., Kodama, Y., Krishna, T., Marinelli, T., Menard, C., Mondelli, A., Muck, T., Naji, O., Nathella, K., Nguyen, H., Nikoleris, N., Olson, L., Orr, M., Pham, B., Prieto, P., Reddy, T., Roelke, A., Samani, M., Sandberg, A., Setoain, J., Shingarov, B., Sinclair, M., Ta, T., Thakur, R., Travaglini, G., Upton, M., Vaish, N., Vougioukas, I., Wang, Z., Wehn, N., Weis, C., Wood, D., Yoon, H., & Zulian, ' (2020). The gem5 Simulator: Version 20.0+. ArXiv, abs/2007.03152.