

CoreWarUCM: Un torneo de programación de virus como apoyo en la adquisición de competencias específicas y potenciador de las competencias transversales del estudiante

José Luis Vázquez Poletti¹ y Juan Carlos Fabero Jiménez²

Resumen: El torneo de programación de virus CoreWarUCM³, una de las actividades que representa anualmente a la Universidad Complutense de Madrid en la Feria Aula en IFEMA, es la parte más visible de una experiencia que sirve de apoyo para los estudiantes de diversas asignaturas impartidas en la Facultad de Informática. En esta actividad se ha resucitado un juego de programación de 1984 donde se debe desarrollar un programa que aniquile al del rival, mientras se trata de dominar el entorno donde se ejecuta. El aprendizaje del lenguaje de programación y la mecánica del enfrentamiento permite al estudiante repasar conceptos clave o prepararle para adquirirlos, en el caso de no haber cursado las asignaturas. La organización y realización del evento CoreWarUCM corre principalmente a cargo de estudiantes voluntarios, los cuales ven potenciada una serie de competencias transversales en el ejercicio de las responsabilidades asumidas.

Palabras clave: Competición; programación; ensamblador; ciberseguridad.

1. CoreWar

“La Guerra en el Núcleo”, término elegido en honor a las primeras memorias de núcleo magnético (como las conservadas en el Museo de la Informática “Profesor José García Santesmases”⁴), enfrenta a dos programas desarrollados en un lenguaje ensamblador llamado Redcode. Aunque fue lanzado en 1984 por A. K. Dewdney y D. G. Jones (Dewdney 1984, 43), éste fue la evolución natural de otro juego de 1960, llamado Darwin, y creado por V. Vyssotsky, R. Morris Sr. y D. Ritchie.

Volviendo al lenguaje, Redcode es una versión reducida de MIPS, con un repertorio consistente en sólo 18 instrucciones más 1 tipo de dato, en el que almacenar los valores necesarios para la ejecución. Con él, el jugador desarrolla un código cuyo programa resultado deberá acabar con el de su contrincante.

Los combates tienen lugar en “el núcleo”, una memoria circular compuesta por un número finito de posiciones (8.000 según el estándar). La posición de inicio de cada

¹ Departamento de Arquitectura de Computadores y Automática. Facultad de Informática. Email: jlvazquez@fdi.ucm.es. ORCID: 0000-0002-6241-8141

² Departamento de Arquitectura de Computadores y Automática. Facultad de Informática. Email: jcfabero@ucm.es. ORCID: 0000-0003-4977-9556

³ <https://corewar.ucm.es/>

⁴ <https://www.fdi.ucm.es/migs/>

programa es aleatoria en cada asalto y se asigna las posiciones contiguas a cada una de las instrucciones que los componen. Un programa se declara vencedor cuando consigue que el rival ejecute una instrucción ilegal (un dato), para lo que deberá modificar las posiciones de memoria ocupadas por sus instrucciones.

El ejemplo más sencillo de programa en RedCode es el "gusano" o "diablillo" (imp):

- MOV 0,1

Esta instrucción hace uso del direccionamiento directo, que es siempre relativo a la posición de la propia instrucción en la memoria. En este caso, el primer operando, 0, hace referencia a la posición de memoria donde se encuentra la instrucción y el segundo operando, 1, se refiere a la posición siguiente. El resultado de la ejecución de esta instrucción es la copia de sí misma a la posición siguiente de memoria. En el siguiente turno de ejecución se ejecuta la instrucción recién copiada, y así sucesivamente. En resumen, este programa tan sencillo se va copiando a sí mismo por la memoria "reptando" por el núcleo.

- MOV 0,1 # inicial
- MOV 0,1 # tras el primer ciclo
- MOV 0,1 # tras el segundo ciclo

Otro ejemplo sencillo es el "enano" (dwarf), que lanza piedras (instrucciones DAT) por la memoria con la esperanza de que alguna de ellas sea ejecutada por el contrincante:

- ADD #4,3
- MOV 2,@2
- JMP -2
- DAT #0,#0

Con este programa los estudiantes aprenden a manejar los direccionamientos inmediato (#) e indirecto (@), además del concepto de bucle:

La instrucción ADD #4,3 suma un valor inmediato, en este caso el 4, al contenido de la posición de memoria situada 3 posiciones más adelante, lo que corresponde con la posición de DAT.

MOV 2,@2 copia el contenido de la posición de memoria +2 (de nuevo el DAT) en la posición de memoria "apuntada" por el contenido del propio DAT (direccionamiento indirecto).

JMP -2 devuelve la ejecución a la posición de memoria -2 desde la actual, es decir, la siguiente instrucción que se ejecute será de nuevo el ADD.

El efecto de este programa es que, en cada ejecución del bucle, va copiando la instrucción DAT en posiciones de memoria cada vez más alejadas, con incrementos de 4 posiciones cada vez. Es decir, va "apedreando" el núcleo con la intención de modificar el código del programa contrincante y que éste acabe ejecutando una de las instrucciones DAT.

Estos son sólo dos ejemplos de partida de los programas que se pueden hacer en RedCode, pero que sirven para ilustrar algunos de los conceptos más importantes en la

programación en cualquier lenguaje ensamblador: los modos de direccionamiento y los bucles.

En los talleres de presentación, además, se incorporan la bifurcaciones condicionales, con las instrucciones JMZ, JMN, DJN, etc., e incluso, la multitarea por medio de la instrucción SPL (split).

2. CoreWarUCM

A continuación se describirá la mecánica actual del torneo para posteriormente, completar con un recorrido histórico que permite entender mejor el formato actual.

2.1. Descripción de la actividad

Este torneo, cuya primera edición se celebró en 2018, consta de 3 fases. Estas son: captación, preparación y competición.

En la fase de captación se realizan talleres de no más de 2 horas de duración orientados a estudiantes de la Facultad de Informática y de bachillerato, enmarcados en actividades más amplias como la Semana de la Ciencia de Madrid o la Semana de la Informática de la propia Facultad. En ellos se emplean 30 minutos en explicar las reglas y enseñar el manejo de A.R.E.S.⁵, el entorno de desarrollo y simulación elegido. Durante el resto del tiempo del taller, los asistentes desarrollan su primer código con la ayuda de los estudiantes voluntarios y veteranos de CoreWarUCM. Además, algunos de estos estudiantes han elaborado documentación de gran interés para los iniciantes, entre la que se encuentra una guía completa (Méndez, 2019) y una ficha de referencia (Méndez, 2020). Por último, los códigos de los asistentes son enfrentados para completar la explicación de la mecánica del torneo.

La fase de preparación comienza cuando se hace pública la fecha del torneo. Para ello se elabora un vídeo de promoción en el que se utilizan elementos de la franquicia de videojuegos llamada Watchdogs⁶, y que está ambientada en el mundo hacker. Todos los años, la creadora de la franquicia (Ubisoft) no sólo concede los permisos de uso de su material para el tráiler, sino que además suministra los premios del torneo.

Para aprovechar el empuje publicitario, se potencia el hecho de que los competidores están desarrollando virus que compiten entre sí para dominar el sistema que están infectando. Es más, el propio A.K. Dewdney se inspiró en la historia del primer virus (Creep) y antivirus (Reaper) cuando desarrolló la primera versión de CoreWar⁷.

Siempre durante la fase de preparación, los estudiantes voluntarios están a disposición de los potenciales competidores para resolver sus dudas a través del correo electrónico.

⁵ <https://harald.ist.org/sites/ares/>

⁶ <https://www.ubisoft.com/es-es/game/watch-dogs/>

⁷ <https://corewar.co.uk/creeper.htm>



Figura 1. Formación ofrecida por estudiantes voluntarios a uno de los institutos de secundaria. Fuente: archivo personal.

El envío de los códigos pone fin a la fase de preparación. Se permiten entregas varias entregas por competidor y en equipo. Después de una validación de los códigos, comprobando que sus instrucciones son correctas y que no son un plagio completo de otros ya existentes (si bien se permite usarlos de punto de partida realizando un mínimo de modificaciones), se emplaza a los competidores al día del torneo.

La fase de competición ha sido presencial en las ediciones de 2018, 2019 y 2020. La primera tuvo lugar en el Vicerrectorado de Estudiantes y las otras dos en IFEMA, durante la Feria Aula. El formato ha sido siempre el mismo: el torneo se presenta como una velada pugilística en la que los desarrolladores cuyos virus se enfrentan en la ronda actual son presentados en el escenario, donde se les va entrevistando a medida que los combates van teniendo lugar.

El espectáculo que supone el torneo se complementa con diferentes elementos. Desde la primera edición se cuenta con un DJ⁸ que va amenizando el evento con una selección de temas de música electrónica, la cual va ajustando acorde con la evolución de los combates. Las ediciones que se celebraron en IFEMA contaron además con un juego de láseres y máquina de humo, que tenían como objetivo principal atraer público, potenciales estudiantes de la UCM, entre los asistentes de la Feria Aula.

La competición en sí comienza con una fase de grupos, los cuales han sido publicitados unos días antes de la competición. De esta fase saldrán los competidores que pasarán a octavos de final (en función del número de inscritos).

En cada ronda se realizan 3 enfrentamientos, declarándose ganador al virus que haya cosechado más victorias. En caso de empate, el jurado (compuesto por profesores y

⁸ <https://corewar.ucm.es/dj-tocreload/>

estudiantes voluntarios) contabilizará las posiciones de memoria ocupadas por cada contendiente en el último enfrentamiento.

Para mantener la atención del público durante las pausas en las que se calculan las siguientes rondas y se introducen los virus en el simulador, se llama al escenario a diversas autoridades de la UCM para que promocionen la institución.

Debido a la pandemia del COVID-19, la edición de 2021 de CoreWarUCM se celebró de forma online por primera vez en su historia⁹. Para ello, se replantearon varios aspectos de la realización para adaptar el torneo al nuevo formato. Como novedad, se colaboró con Banco de Alimentos de Madrid¹⁰, obteniendo al final del torneo casi 400Kg de alimentos.

La retransmisión se realizó a través del canal de YouTube de la UCM y se enlazó desde la plataforma digital de la Feria Aula en IFEMA. Los presentadores, el jurado y el equipo de realización estaban físicamente en la Sala de Juntas de la Facultad de Informática, cumpliendo con las medidas de protección sanitaria. Los competidores, el DJ y los invitados especiales fueron entrando online desde sus respectivas ubicaciones.

Para la realización de la transmisión se utilizó Open Broadcaster Software¹¹, una solución gratuita y de código abierto. A cada participante se le suministraba una dirección generada a través del servicio gratuito OBS.ninja, la cual capturaba su webcam y micrófono a través del navegador web, sin necesidad de realizar ninguna instalación adicional. La coordinación se realizaba a través de Google Meet.



Figura 2. Desarrollo de CoreWarUCM 2021. Fuente: archivo personal.

⁹ <https://www.youtube.com/watch?v=7wAa5F8xZQ0>

¹⁰ <https://bamadrid.org/>

¹¹ <https://obsproject.com/>

2.2. Evolución histórica

No se podría entender la organización de CoreWarUCM en su forma actual sin realizar un recorrido histórico desde la introducción de CoreWar en la docencia de la Facultad de Informática.

Dicha introducción fue realizada por los autores de la presente contribución en la asignatura Redes y Seguridad, perteneciente al Grado en Ingeniería Informática, durante el curso 2015/2016. La motivación fue la inexistencia de prácticas de laboratorio relacionadas con el desarrollo de malware. CoRyS War, el nombre que recibió la actividad, constituyó para los estudiantes una forma sencilla y divertida de estudiar, implementar y mejorar las estrategias de las diferentes amenazas malware existentes. Los estudiantes eran divididos en dos equipos (tantos como grupos de la asignatura) y enviaban un código que les representara antes de una fecha límite. A continuación se producía un vídeo con el resumen de los enfrentamientos, mezclado con escenas de la película Tron: Legacy¹², que se subiría al canal de YouTube de la Facultad de Informática y se mostraría en las pantallas del edificio, al más puro estilo resumen de la jornada deportiva.

En 2016 se celebró CoreWarRITSI¹³ durante el VII Congreso de esta asociación nacional que aglutina estudiantes de todas las ramas de Informática. En el torneo participaron 23 estudiantes de la Universidad Complutense de Madrid, la Universidad Autónoma de Madrid y la Universidad de Granada. Como preparación para este evento, se celebró en la Facultad de Informática la primera CoreWarFDI¹⁴, con el objetivo de preparar a los estudiantes propios. Para ello se contó con la presencia física y los consejos de Neogryzor, el campeón español de la liga internacional de CoreWar.



Figura 3. Desarrollo de CoreWarRITSI (2016). Fuente: archivo personal.

¹² <https://www.imdb.com/title/tt1104001/>

¹³ <https://madrid2016.congreso.ritsi.org/corewar/>

¹⁴ <http://bit.ly/CoreWarFDI>

En 2018 se celebró la primera CoreWarUCM. Tal y como se indicó anteriormente, ésta tuvo lugar en el Vicerrectorado de Estudiantes. Participaron 12 competidores y como hecho notable, tres de los virus presentados fueron desarrollados por el equipo de ciberseguridad de la empresa farmacéutica Roche. Dicho equipo aprovechó su participación para dar a conocer las oportunidades de trabajo a los estudiantes.

Fue en 2019 cuando la primera CoreWarUCM tuvo lugar en el recinto ferial IFEMA, durante la Feria Aula y en ella compitieron 16 desarrolladores. Al año siguiente, esta cifra subió a 22, registrándose por primera vez la participación de estudiantes de bachillerato. La última edición (online) reunió a 22 competidores y fue la primera en la que se colaboró con Banco de Alimentos de Madrid.



Figura 4. Desarrollo de CoreWarUCM 2020. Fuente: archivo personal.

3. Competencias desarrolladas

En la evaluación de las competencias adquiridas por los participantes de CoreWarUCM hemos de distinguir dos tipos de participación. Por un lado, se encuentran los estudiantes que colaboran de forma activa en la organización del evento. Por otro lado están los competidores, que desarrollan los códigos que se enfrentan en el torneo.

Para el primer grupo, las competencias van más allá de las meramente académicas. Se hacen responsables de algunos aspectos organizativos del evento, lo que les lleva a buscar patrocinadores, organizar intendencia, buscar espacios donde se desarrollen las distintas fases del concurso... Todo esto les hace ser proactivos, aumentar la confianza en sí mismos al sentirse libres de proponer sus ideas y, además, les ayuda a acrecentar su creatividad en la búsqueda de nuevas formas de motivar al resto de participantes.

De hecho, una de las señas de identidad del evento CoreWarUCM es la ambientación que se otorga cada año, tal y como se explica en la Sección 1.2. Los estudiantes que participan en la organización ponen en marcha su imaginación para la creación de un ambiente que resulte atractivo al resto de participantes y público.

En cuanto a los competidores, el hecho de tener que plasmar una determinada estrategia para vencer al contrario mediante el uso de un conjunto reducido de instrucciones máquina les ayuda a ser capaces de analizar un problema complejo y dividirlo en una

serie de pasos simples. Esto es de vital importancia en asignaturas que tengan que ver con la algoritmia, como Fundamentos de la Programación, pero también con aquellas que están más relacionadas con el hardware y el lenguaje ensamblador, como Fundamentos de Computadores II y Estructura de Computadores. A lo largo de estos años se ha recibido una respuesta muy positiva de los estudiantes de dichas asignaturas que participaron en la actividad, indicando una mayor facilidad a la hora de enfrentarse a éstas.

Por ejemplo, en la asignatura de Fundamentos de Computadores II, de 1er curso, en el Módulo I se describen las instrucciones básicas de un microprocesador ARM y se hace una introducción al lenguaje ensamblador. Igualmente, en la asignatura Estructura de Computadores, de 2º, también se emplea el lenguaje ensamblador. Aunque la sintaxis de las instrucciones es distinta, la semántica es muy similar. La programación en Redcode facilita a los estudiantes la asimilación de conceptos como las bifurcaciones (saltos), modos de direccionamiento (inmediato, directo e indexado)... y todo ello de una manera lúdica.

Aunque ya se ha indicado en la Sección 2.3, conviene recordar que entre los beneficiados de esta actividad se encuentran los estudiantes de asignaturas relacionadas con la seguridad informática. Destacan Redes y Seguridad, así como Ciberseguridad en Videojuegos, las cuales cuentan con un tema de Seguridad de Sistemas, donde se abordan las amenazas malware.

Por último pero no menos importante, cabe destacar la promoción tanto de la Facultad de Informática como de la Universidad Complutense, ya que el torneo y la fase de captación realizada con institutos están teniendo una buena acogida a lo largo de estos años.

Referencias bibliográficas

Dewdney, Alexander Keewatin. 1984. “Recreational Mathematics–Core Wars”. *Scientific American*. 2: 43-98.

Méndez, Fernando. 2019. “El manual básico de CoreWar: aprende Recode DAT cero”. Última modificación el 19 de febrero de 2019. <https://fdist.ucm.es/corewar/CoreWar.pdf>

Méndez, Fernando. 2020. “REDCODE: Aprende a programar en CoreWar”. Última modificación el 20 de enero de 2020. https://fdist.ucm.es/corewar/Corewar_REDCODE_on_one_page.pdf