



Revisión de MiComputer

18/09/2025

321058526 - Gabriel Hernandez Yukioayax Canek

321152662 - Nochebuena Sosa Kevin Israel

Facultad de Ingeniería, UNAM

Revisión de MiComputer: Fascículo 22


Este fascículo de MiComputer ofrece un artículo técnico sobre el Commodore PET con piezas divulgativas que ayudan a entender conceptos fundamentales de la computación, como la máquina de Turing, la revista parece pertenecer a la época de los ochenta y aunque el contexto tecnológico es distinto, los problemas que aborda son los mismos que estudiamos a día de hoy en Sistemas Operativos, por ejemplo, cómo inicia una computadora, cómo se organiza la memoria y el almacenamiento, cómo se controla el hardware y cómo se estructura el programa que coordina todo, etc.

La sección que aborda el tema de Commodore PET repasa por qué este microordenador fue importante en el mercado doméstico y educativo, pues al encender la máquina, aparecía en pantalla el intérprete de BASIC grabado en ROM listo para aceptar órdenes, este diseño es importante vista desde el punto de los sistemas, ya que el sistema operativo básico vivía en memoria de solo lectura por lo que el arranque era inmediato y fiable, el entorno de uso era monousuario y monoprogramado, es decir, el usuario interactuaba directamente con el intérprete y a través de instrucciones de alto nivel controlaba dispositivos como el “cassette” integrado, el teclado, la pantalla y las interfaces externas.

La lectura también cuenta como estaba armado el equipo, contaba con un CPU 6502, RAM modesta pero ampliable, ROM con el monitor/BASIC, generador de caracteres con mayúsculas y minúsculas, y el puerto IEEE-488 que era para conectar periféricos de almacenamiento o laboratorio, también menciona el uso de cintas como medio principal y la posibilidad de usar disquetes en modelos o interfaces compatibles.

En la práctica, el PET era una máquina en donde el software y hardware estaban muy cerca, pues era posible leer y escribir posiciones de memoria con PEEK/POKE y acceder a registros de E/S sin capas de protección.

En sistemas operativos el PET era un diseño muy sencillo y transparente, pues al arrancar desde la ROM no necesitaba depender de un disco, no había cuentas de usuario ni permisos especiales y tampoco un núcleo que aislara procesos porque en realidad no tenía multitarea, el programa del usuario y el sistema compartían la misma memoria, lo que hacía que fuera fácil de aprender y experimentar pero también significaba que cualquier error podía colgar toda la máquina y no había forma de proteger datos o separar tareas, el PET prácticamente no ofrecía niveles de privilegio ni separación entre usuario y núcleo, todo el código corría con la misma autoridad lo que hoy conoceríamos como una superficie de riesgo amplia, esto contrasta con los sistemas modernos que aíslan memoria, procesos y dispositivos, y que ejecutan al usuario en modo no privilegiado.



Por otro lado, la persistencia se lograba en cassette o disquete, en el artículo se asume medios sin ningún tipo de cifrado ni control de integridad que, vista desde los sistemas operativos modernos, esto se relaciona con los subsistemas de archivos, el journaling y las capas de cifrado a nivel de bloque o de fichero que hoy consideramos algo normal. En la sección “La cinta teórica” se presenta la máquina de Turing como un dispositivo idealizado para razonar sobre algoritmos y sobre la pregunta de si un problema es informatizable, la explicación parte de tres componentes:

- Una cinta potencialmente infinita que almacena símbolos.
- Una cabeza que lee y escribe moviéndose a izquierda o derecha.
- Una unidad de control que selecciona a partir del estado actual y del símbolo observado la acción a ejecutar.

El artículo muestra el formato de quintuplos y construye un ejemplo concreto para calcular una función lógica AND paso a paso sobre la cinta.

Esto es importante porque el modelo destila la esencia del cómputo secuencial (estado, memoria, E/S y control); un kernel real no es una máquina de Turing pero su bucle principal encarna el mismo principio el cual es leer un evento, decidir una acción según reglas y estados, escribir cambios y avanzar, la cinta sugiere además una visión unificada de memoria y almacenamiento secuencial que enlaza con los cassettes del PET y con la noción de archivo.

Cuando hablamos de aislamiento y de planificación, el modelo ayuda a no perder la perspectiva, si todo es una transición de estados sobre símbolos entonces aislar procesos significa mantener estados independientes y evitar interferencias en la cinta de cada uno, planificar es decidir qué cabeza avanza en cada instante y la persistencia es garantizar que ciertas marcas de la cinta sobreviven a la pérdida de energía.

En conclusión, el fascículo presenta temas que son fundamentales con sistemas operativos y tienen relación con lo visto en clase, el caso del Commodore PET muestra un sistema simple que permite entender por contraste la necesidad de privilegios, aislamiento y seguridad en los sistemas operativos modernos mientras que la máquina de Turing aporta un modelo conceptual que explica el cómputo como una secuencia de estados y transiciones, lo que conecta con la gestión de memoria, procesos y persistencia, en conjunto, ambos apartados complementan lo estudiado y refuerzan cómo las bases históricas siguen siendo esenciales en la comprensión actual de los sistemas operativos.

Referencias:

- Commodore Business Machines, Inc. (1978). *PET 2001-8 personal computer: User manual*. Commodore Business Machines.
[https://www.masswerk.at/pet/manuals/PET_User_Manual_\(2001-8\).pdf](https://www.masswerk.at/pet/manuals/PET_User_Manual_(2001-8).pdf)
- Primrose, B. (comp.) (1980). *PET and the IEEE-488 Bus*. (Manual de interfaz IEEE-488 para PET).
https://ia601802.us.archive.org/17/items/PET_and_the_IEEE488_Bus/PET_and_the_IEEE488_Bus_text.pdf
- ICS Electronics. (2018, 23 de marzo). *GPIB 101: A tutorial about the GPIB bus*.
https://www.icselect.com/pdfs/ab48_11%20GPIB-101.pdf
- Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), 230–265; 2(43), 544–546 (corr.).
https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf