Memoria virtual

El concepto de la **memoria virtual** es que es el método para conseguir que la suma de los espacios de pila, datos y texto de un programa pueda ser mayor que el tamaño físico de la memoria disponible para él. El S.O. mantiene en memoria únicamente las partes del programa que se están utilizando y mantiene en disco el resto. Denominamos **conjunto residente** a la parte del proceso que está en la memoria principal.

La memoria virtual permite optimizar el uso de memoria principal, mantiene ‘**más procesos**’ en memoria principal albergando solo una parte (suficiente) de cada uno de ellos, permite que el proceso sea más grande que toda la memoria principal.

La memoria virtual tiene problemas tales como:

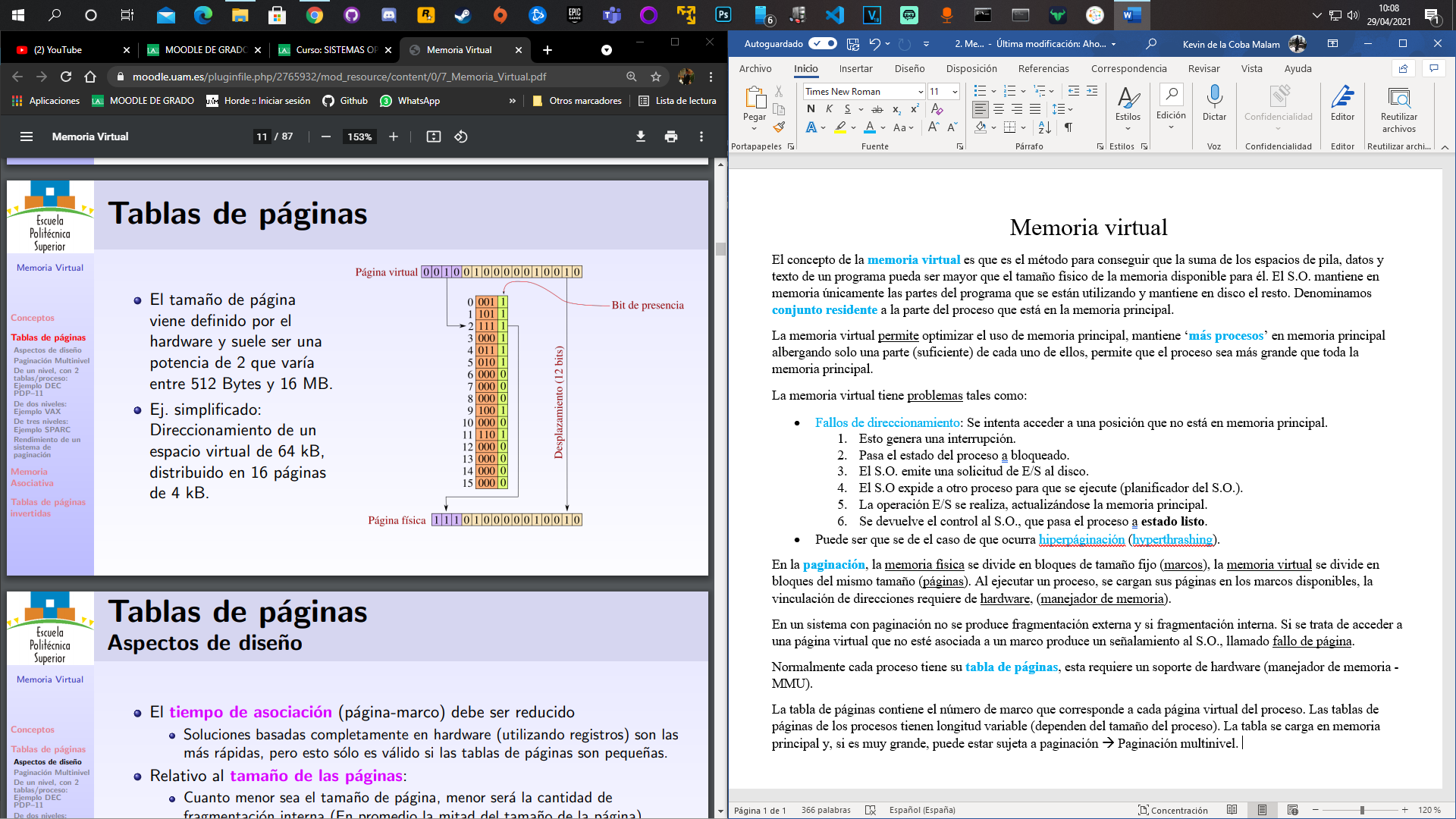
* Fallos de direccionamiento: Se intenta acceder a una posición que no está en memoria principal.
  1. Esto genera una interrupción.
  2. Pasa el estado del proceso a bloqueado.
  3. El S.O. emite una solicitud de E/S al disco.
  4. El S.O expide a otro proceso para que se ejecute (planificador del S.O.).
  5. La operación E/S se realiza, actualizándose la memoria principal.
  6. Se devuelve el control al S.O., que pasa el proceso a **estado listo**.
* Puede ser que se de el caso de que ocurra hiperpáginación (hyperthrashing).

En la **paginación**, la memoria física se divide en bloques de tamaño fijo (marcos), la memoria virtual se divide en bloques del mismo tamaño (páginas). Al ejecutar un proceso, se cargan sus páginas en los marcos disponibles, la vinculación de direcciones requiere de hardware, (manejador de memoria).

En un sistema con paginación no se produce fragmentación externa y si fragmentación interna. Si se trata de acceder a una página virtual que no esté asociada a un marco produce un señalamiento al S.O., llamado fallo de página.

Normalmente cada proceso tiene su **tabla de páginas**, esta requiere un soporte de hardware (manejador de memoria - MMU).

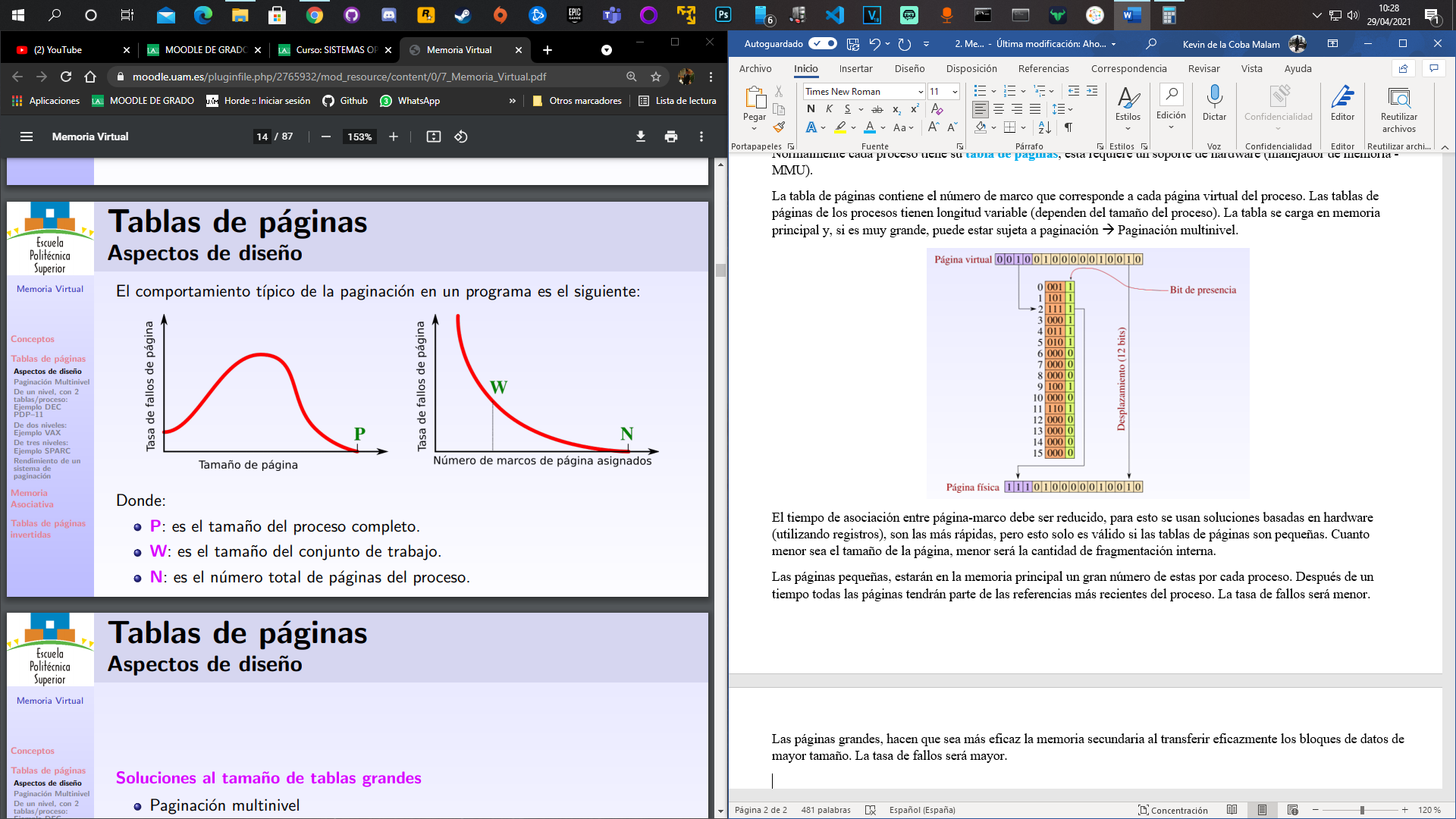
La tabla de páginas contiene el número de marco que corresponde a cada página virtual del proceso. Las tablas de páginas de los procesos tienen longitud variable (dependen del tamaño del proceso). La tabla se carga en memoria principal y, si es muy grande, puede estar sujeta a paginación 🡪 Paginación multinivel.



El tiempo de asociación entre página-marco debe ser reducido, para esto se usan soluciones basadas en hardware (utilizando registros), son las más rápidas, pero esto solo es válido si las tablas de páginas son pequeñas. Cuanto menor sea el tamaño de la página, menor será la cantidad de fragmentación interna.

Las páginas pequeñas, estarán en la memoria principal un gran número de estas por cada proceso. Después de un tiempo todas las páginas tendrán parte de las referencias más recientes del proceso. La tasa de fallos será menor.

Las páginas grandes, hacen que sea más eficaz la memoria secundaria al transferir eficazmente los bloques de datos de mayor tamaño. La tasa de fallos será mayor.



Para tratar las páginas grandes, se usa la paginación multinivel o las tablas de páginas invertidas.

La **paginación multinivel**:

