

Spec 2000

Spec (empresa) cada ciertos años (2000, ...) hace un conjunto de instrucciones para hacer benchmark de los diferentes procesadores con diversos tamaños de caché.

Benchmarking

```
tasa de fallos = #fallos / #referencias
```

```
tasa de aciertos = #aciertos / #referencias
```

La tasa de fallos depende del programa.

(¡IMPORTANTE!) TSA: Tiempo de Servicio en caso de Acierto ---> cuanto tarda la caché en dar un dato que esté en la caché.

(¡IMPORTANTE!) TSA: Tiempo de Servicio en caso de Fallo ---> cuanto tarda la caché en dar un dato que **NO** está en la caché.

El tamaño importa. Cuanto más grande sea la caché, mejor. Pero claro, cuanto mayor es la caché, más grande es su TSA (tarda más).

Cuanto mayor asociatividad, mejor. Pero claro, cuanta más asociatividad tiene la caché, más grande es su TSA (tarda más).

Por lo tanto, no existe una caché perfecta, unas irán bien para unas cosas y otras irán bien para otras. Lo que hacemos para escoger bien una caché es simular millones de instrucciones con tamaños diferentes de caché y coger la que da mejores resultados.

```
tiempo medio de acceso = hits * TSA + miss * TSF
```

o, lo que es lo mismo:

```
tiempo medio de acceso = TSA + miss * penalización
```

Dependiendo de la jerarquía de memoria, las fórmulas cambiarán.

```
tiempo de ejecución = #instrucciones * CPI * tiempo de ciclo  
// tiempo de ciclo = 1 / frecuencia
```

El CPI ideal te lo da el fabricante. El CPI real es:

```
CPI = #referencias * (tiempo medio de acceso - TSA)
```

O, lo que es lo mismo:

```
CPI = #referencias * miss * penalización
```

Memoria Virtual

Necesito un mecanismo que me traduzca la dirección 0x000... del proceso 1 a la dirección X de memoria física. Porque sinó, con la dirección 0x000... del proceso 2, se solaparía con la del otro proceso.

Paginación

La paginación consiste en que yo cojo la memoria principal y la divido en trozos iguales que se llaman "marcos de página". Entonces para cada página de un proceso le asigno un marco de página. Entonces, en vez de cargar todo el proceso, solamente cargo aquella parte del proceso que necesito.

La página 6 del proceso P2 corresponde a la página 4 de memoria principal.

(páginas virtuales) --> corresponden a --> (páginas física)

DISCO -- página --> RAM -- bloque --> CACHÉ -- dato --> CPU

La memoria virtual es completamente asociativa (+ LRU). El sistema operativo tiene que pensárselo muy bien antes de reemplazar una página.

PÁGINA	DESPLAZAMIENTO

El número de la página
(este número se traduce
para ir de MV a MF)

Los bits de desplazamiento de la página (se mantienen
constantes en la traducción). Tiene tantos bits como el
tamaño de la página (4KB --> 12 bits)

El tamaño de la tabla de páginas es de 2^{52} filas (con páginas de 4KB).

TLB

Es una caché de la tabla de páginas para acceder más rápido y no tener que ir a buscar a la memoria física.

Segmentación

- **DS (DATA SEGMENT)**
- **SS (STACK SEGMENT)**
- **IS (INSTRUCTION SEGMENT)**

Para acceder a una dirección de memoria es `segmento + offset de esa sección`.

Si quiero acceder a la 70 del segmento de datos hago: `70 + DS`