

Έστω σύστημα με ένα επίπεδο κρυφής μνήμης (δηλ. έχουμε ΚΜΕ – κρυφή μνήμη – κύρια μνήμη).

Έστω επίσης ότι η κρυφή μνήμη έχει 1024 γραμμές και κάθε γραμμή χωράει 64 bytes.

Ποια η χωρητικότητα της κρυφής μνήμης;
 $1024 * 64 \text{ bytes} = 64\text{KB}$

Πόσοι C ints και πόσοι C double χωράνε σε κάθε γραμμή;
 $64/4 = 16 \text{ ints}$ ή $64/8 = 8 \text{ doubles}$

Έστω ότι η πολιτική τοποθέτησης στην κρυφή μνήμη είναι η εξής:

Τα bytes με διευθύνσεις 0..63 πάνε στη γραμμή 0 ($0*64 \dots 0*64+63$)
Τα bytes με διευθύνσεις 64..127 πάνε στη γραμμή 1 ($1*64 \dots 1*64+63$)
Τα bytes με διευθύνσεις 128..191 πάνε στη γραμμή 2 ($2*64 \dots 2*64+63$)
κοκ...

Ποιες διευθύνσεις τοποθετούνται στην τελευταία γραμμή (γραμμή 1023);
($1023*64 \dots 1023*64+63$) ($1023*64=65472 \dots 65472+63=65535$)

Στη συνέχεια η τοποθέτηση επαναλαμβάνεται από την γραμμή 0 ξανά.
Δηλαδή, τα bytes με διευθύνσεις 65536 ($1024*64$)...65599 πάνε στη γραμμή 0
Τα bytes με διευθύνσεις 65600..65663 πάνε στη γραμμή 1
65664.. (65668) ... $65664+63$ στη γραμμή 2
και αυτό επαναλαμβάνεται συνεχώς.

Έστω ένα πρόγραμμα C

```
int i = 33; // i = 0x21;
```

```
int main() {  
...  
  i++;  
...  
}
```

Πόσα bytes στη μνήμη καταλαμβάνει το i;
4 bytes (C int)

Ποια η τιμή των bytes αυτών; 0x 00 00 00 21

Υποθέστε ότι την αρχικοποίηση την κάνει ο μεταγλωττιστής (το 33 είναι μέσα στα αρχικά δεδομένα του προγράμματος). Αν η αποθήκευση γίνεται στις θέσεις 65668..65671 και το σύστημα είναι little-endian, τι περιέχει καθένα από τα bytes αυτά;

$65668 = 21$
 $65669 = 00$
 $65670 = 00$
 $65671 = 00$

Κατά την εκτέλεση του main() πόσες και ποιες προσπελάσεις θα κάνει η ΚΜΕ στο σύστημα μνήμης για το i++;

$i = i + 1;$

πρέπει να διαβάσω από τη μνήμη το i , να το αυξήσω κατά 1, και να γράψω πίσω στη μνήμη τη νέα τιμή.

2 προσπελάσεις (1 ανάγνωση εύρους 4 bytes/1 εγγραφή εύρους 4 bytes) στη θέση 65668 και στις 3 επόμενες.

Αλληλεπίδραση με την κρυφή μνήμη του συστήματος.

KME: (προς cache) read mem[65668]/4 bytes

Cache: (δεν έχει τα δεδομένα αυτής της διεύθυνσης)

Cache: (θα το ζητήσει από την κύρια μνήμη)

Cache: (προς κύρια μνήμη) Ανάγνωση 64 bytes δεδομένων από τις διευθύνσεις 65664 έως και 65664+63 (=65727)

Κύρια μνήμη: (προς cache) αποστολή του ζητούμενου block με 64 bytes δεδομένων

Cache: (τοποθετεί το block στη γραμμή 2)

Cache: (προς KME) αποστολή των 4 bytes του i που ζητήθηκαν (από τις διευθύνσεις 65668...65671)

KME: αυξάνει το i κατά 1

KME: (προς cache) write mem[65668]/4 bytes

Cache: (έχει το μπλοκ στη γραμμή 2)

Cache: (τοποθετεί τη νέα τιμή του i που γράφει η KME στην κατάλληλη θέση της γραμμής 2)

Σημείωση 1: Στο σημείο αυτό η κύρια μνήμη ΔΕΝ έχει τη νέα τιμή του i !

Σημείωση 2: Πώς ξέρει η κρυφή μνήμη ότι στη γραμμή 2 είναι το μπλοκ με διευθύνσεις 65664..65727 κι όχι π.χ. το μπλοκ 128..191;

Υπάρχει μια ετικέτα (tag) σε κάθε γραμμή της κρυφής μνήμης που λέει: «εδώ είναι το μπλοκ 65664..65727». Η κρυφή μνήμη συγκρίνει το tag με αυτό που αναζητά.

Άλλο παράδειγμα.

Έστω ότι αν η ΚΜΕ βρει αυτό που θέλει στην κρυφή μνήμη η προσπέλαση κοστίζει 2 κύκλους ρολογιού. Αν όχι (και πρέπει να έρθει από την κύρια μνήμη), η προσπέλαση κοστίζει 100 κύκλους ρολογιού.

Ένα πρόγραμμα C είναι ως εξής:

```
int mat[992]; // έστω ότι τοποθετείται στη μνήμη στις θέσεις με διεύθυνση 640 και πάνω
```

```
int main() {  
...  
int sum = 0;  
for (i=0;i<992;i++) {  
    sum += mat[i]; // sum = sum + mat[i]  
}  
  
...  
}
```

Πόσο θα σας κοστίσουν σε κύκλους οι προσπελάσεις στον πίνακα mat;

Τοποθέτηση ανά 16 ακεραίους σε γραμμές της κρυφής μνήμης. Ο πίνακας έχει 62 ολόκληρες γραμμές κρυφής μνήμης.

mat[0] Ζητάτε τις διευθύνσεις 640...643. Δεν υπάρχουν στην κρυφή μνήμη. **100 κύκλοι**
mat[1]..mat[15]. Υπάρχουν στην κρυφή μνήμη. $2 \times 15 = 30$ **κύκλοι**.
Συνολικά για την 16άδα = **130 κύκλοι**.
Για όλες τις 16άδες = $62 \times 130 = 8060$ **κύκλοι ρολογιού**.

Αν δεν είχατε την κρυφή μνήμη και κάθε ανάγνωση ακεραίου κόστιζε όπως πριν 100 κύκλους, ποιο θα ήταν το κόστος της προσπέλασης του mat;
 $992 \times 100 = 99200$ **κύκλοι!**