CS 354 - Machine Organization & Programming
Tuesday, October 15, 2019

Project p3 (6%): DUE at 10 pm on Monday, October 28th

Homework hw3 (1.5%): DUE at 10 pm on Friday, October 18th

Last Time

Placement Policies
Free Block - Too Large/Too Small
Coalescing Free Blocks
Footers

Today

Footers (from last time)
Explicit Free List (from last time)
Explicit Free List Improvements
Heap Caveats
Memory Hierarchy
Locality

Next Time

Designing Caches
Read: B&O 6.4 intro - 6.4.2

Copyright © 2016-2019 Jim Skrentny

Explicit Free List Improvements

Free List Ordering

address order:

malloc with FF

free

last-in order:

malloc with FF

free

Free List Segregation

simple segregation:

structure

malloc

if list is empty:

free

problem

fitted segregation:

fitting

splitting

coalescing

Copyright © 2016-2019 Jim Skrentny

Heap Caveats

Don’t assume consecutive heap allocations result in contiguous payloads!

→ Why?

Don’t assume heap memory is initialized to 0!

Do free all heap memory that your program allocates!

→ Why are memory leaks bad?

→ Do memory leaks persist when a program ends?

Don’t free heap memory more than once!

→ What is the best way to avoid this mistake?

Don’t read/write data in freed heap blocks!

→ What kind of error will result?

Don’t change heap memory outside of your payload!

→ Why?

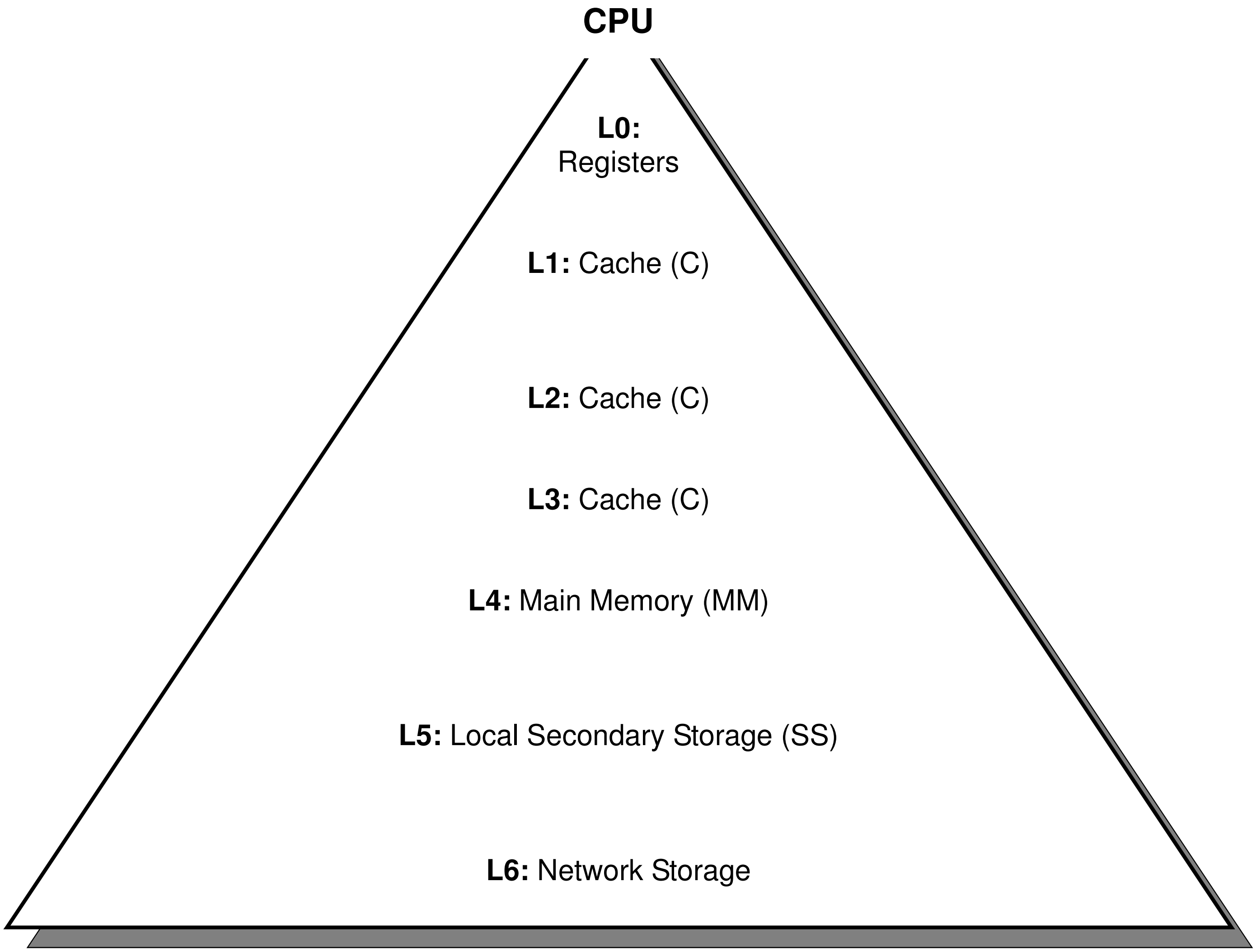
Do check if your memory intensive program has run out of heap memory!

→ How?

Copyright © 2016-2019 Jim Skrentny

Memory Hierarchy

✵ *The memory hierarchy*



CPU

L0: Registers

L1: Cache (C)

L2: Cache (C)

L3: Cache (C)

L4: Main Memory (MM)

L5: Local Secondary Storage (SS)

L6: Network Storage

Cache

Memory Units

word: size used by transfer between

block: size used by transfer between

page: size used by transfer between

Memory Transfer Time

cpu cycles:

latency:

Copyright © 2016-2019 Jim Skrentny

Locality

✵ *Programs with good locality*

Why? Programs with good locality

What?

<u>temporal locality</u>: when a recently accessed memory location

<u>spatial locality</u>: when a recently accessed memory location

Example

```
int sumArray(int a[], int size, int step) {
   int sum = 0;
   for (int i = 0; i < size; i += step)
      sum += a[i];
   return sum;
}
```

→ List the variables that demonstrate temporal locality.

→ List the variables that demonstrate spatial locality.

<u>stride</u>:

Copyright © 2016-2019 Jim Skrentny