

Aluna: Thaisa Mirely

Descreva a arquitetura do seu computador pessoal em termos dos seguintes itens:

- Frequência de clock.

**R: [1] e [2]**

2.4 GHz

- Número de núcleos (físicos e virtuais) do processador.

**R: [1] e [2]**

Um processador com dois núcleos físicos. [Intel Core 2 Duo P8600]

- Qual é a organização da memória? Ela é NUMA ou não? COMA? Barramento?

**R: [11]**

*symmetric multiprocessing (SMP)* utilizando barramento.

- Número de níveis de cache e onde as memórias cache estão localizadas.

**R: [1]**

-L1: 2 x 32 KB 8-way set associative instruction caches

2 x 32 KB 8-way set associative write-back data caches

-L2: 3 MB 12-way set associative shared cache

- Como é mantida a coerência das memórias cache.

**R: [7]**

Ambos Mac OS 9 e Mac OS X acomodar multi-processamento simétrico (SMP), que trata todos os processadores de forma igual. Utilizando OS X Yosemite há o mesmo protocolo de controle.

(SMP definição: múltiplos processadores similares em um mesmo computador, conectados por um barramento.)

- As instruções não triviais (por exemplo, um LOAD ou um STOR são instruções "triviais" por fazerem apenas uma coisa) que estão disponíveis e o que elas fazem.

**R: [8], [9] e [10]**

**compare\_and\_swap:** é uma instrução usada em multithreading para conseguir sincronização.

Exemplo:

```

int compare_and_swap(int* reg, int oldval, int newval)
{
    ATOMIC();
    int old_reg_val = *reg;
    if (old_reg_val == oldval)
        *reg = newval;
    END_ATOMIC();
    return old_reg_val;
}

```

**feach\_and\_add:** é uma instrução que modifica o conteúdo de uma localização de memória.

Exemplo:

```

function FetchAndAdd(address location, int inc) {
    int value := *location
    *location := value + inc
    return value
}

```

- A velocidade da memória que está rodando na sua máquina. Ela é rápida o suficiente para o processador? Sim? Não? Por quê?

**R: [4] e [5]**

A velocidade é de 2GB de 1066 MHz DDR3 SDRAM(PC3-8500) instalada em par(dois módulos de 1GB). Levando em conta que o processador passa de 2GHz e há apenas 2GB de memória um opção para aumentar a eficiência seria aumentar o número de memória para 4GB adicionando 2GB ou adicionar mais 2GHz de clock no processador. Como memória tem um custo mais reduzido que um processador, essa opção tornasse mais válida para aplicabilidade. Pelas aplicações executadas no computador se serem moderadamente leves a usabilidade não posta em baixa, executando de forma normal cada aplicação. Algo que seria interessante ser adicionado ao computador seria um SSD para agilizar a inicialização do SO e de demais funções ou troca de memória para um híbrida(HD e SSD)que cria cache dos arquivos e programas mais acessados, o que melhora a performance geral do sistema após algum tempo de uso.

Extra[6]:

## **Breve Resumo de Concorrência em iOS.**

### *Threads and Concurrency*

*The system creates your app's main thread and you can create additional threads, as needed, to perform other tasks. For iOS apps, the preferred technique is to use Grand Central Dispatch (GCD), operation objects, and other asynchronous programming interfaces rather than creating and managing threads yourself. Technologies such as GCD let you define the work you want to do and the order you want to do it in, but let the system decide how best to execute that work on the available CPUs. Letting the system handle the thread management simplifies the code you must write, makes it easier to ensure the correctness of that code, and offers better overall performance.*

*When thinking about threads and concurrency, consider the following:*

- *Work involving views, Core Animation, and many other UIKit classes usually must occur on the app's main thread. There are some exceptions to this rule—for example, image-based manipulations can often occur on background threads—but when in doubt, assume that work needs to happen on the main thread.*
- *Lengthy tasks (or potentially length tasks) should always be performed on a background thread. Any tasks involving network access, file access, or large amounts of data processing should all be performed asynchronously using GCD or operation objects.*
- *At launch time, move tasks off the main thread whenever possible. At launch time, your app should use the available time to set up its user interface as quickly as possible. Only tasks that contribute to setting up the user interface should be performed on the main thread. All other tasks should be executed*

*asynchronously, with the results displayed to the user as soon as they are ready.*

## **Referências:**

- [1] Intel Core 2 Duo P8600 specifications. Disponível em:  
<<http://goo.gl/6IBNmD>>
- [2] Apple MacBook "Core 2 Duo" 2.4 13" (Mid-2010) Specs. Disponível em:  
<<http://goo.gl/B7YwpB>>
- [3] Coerência de Cache. Disponível em: < <http://goo.gl/Bjuu4l> >
- [4] Processador: desvendando o mistério do clock e da velocidade real.  
Disponível em: <<http://goo.gl/xB2JQR>>
- [5] Processor Speed vs Memory. Disponível em: <<http://goo.gl/Z5aJJg>>
- [6] App Programming Guide From iOS. Disponível em:  
<<http://goo.gl/wIycWA>>
- [7] Processors. Disponível em: <<http://goo.gl/Ljl9pb>>
- [8] Intel® 64 and IA-32 Architectures Software Developers Manual.  
Disponível em: <<http://goo.gl/ixkVno>>
- [9] Compare-and-swap. Disponível em: <<http://goo.gl/kFKMr0>>
- [10] Fetch-and-add. Disponível em: <<http://goo.gl/JEum3t>>
- [11] Kernel Programming Guide. Disponível em: <<http://goo.gl/46u62k>>