

6. (6 valores)

Considere uma cache do tipo 8-way set associative com 1024k byte de RAM de dados em linhas de 32 byte, integrada num sistema com endereçamento a 32 bit.

- [2] Determine, em bits, a dimensão dos campos OFFSET, INDEX (SET) e TAG.
- [2] Descreva uma possível política de substituição que seja eficiente.
- [2] A taxa de ocupação de uma cache com 4-way set associative, com a mesma dimensão de memória de dados, é menor que a de uma 8-way set associative. Indique um cenário que confirme esta afirmação.

5. [4 valores]

Escreva, em assembly IA-32, o código da função:

```
unsigned int strspn(const char * str, const char * strCharSet);
```

que retorna o índice do primeiro carácter de str que não ocorre em strCharSet.
Exemplo: strspn("acabada", "abc") deve retornar 5.

6. [3 valores]

Escreva, em assembly IA-32, o código da função

```
const char * longest_match(const char * vstr[], unsigned int nstrs, const char * strCharSet);
```

que recebe, em vstr, um array de nstrs strings e retorna a que começar com o maior número de caracteres presentes em strCharSet.

Exemplo: sendo const char * palavras[] = { "acabei", "abacate", "abadessa" }, a execução de longest_match(palavras, 3, "abc") deve retornar o endereço de "abacate".

Tire partido da função desenvolvida em 5.

4. [8 valores]

No "Intel® 64 and IA-32 Architectures Optimization Reference Manual" (Nov. 06, pp. 3-61) afirma-se que: Nos processadores Intel Core 2 Duo, Intel Core Duo, Intel Core Solo e Pentium M, a utilização simultânea de mais de 8 localizações afastadas entre si de múltiplos de 4 Kbytes resultará num excesso de ocorrências de cache-misses no primeiro nível de cache.

Sabendo que a dimensão de cada linha de cache é de 64 bytes:

- [2] Qual o tipo de organização da cache? Justifique.
- [3] Qual a capacidade (em dados) da cache? Apresente os cálculos.
- [3] Considerando um espaço de endereçamento a 32 bits, qual a capacidade da RAM de tags? Apresente os cálculos.

8. [9 valores]

a) [5] Implemente, em assembly IA-32, a seguinte função:

```
char *struprnt(char * str, unsigned int * pcnt);
```

que converte para maiúsculas os caracteres da string str, deixando na localização apontada por pcnt o número de caracteres convertidos. A função retorna o endereço da sequência convertida. Para a conversão de cada carácter, utilize a função toupper, que recebe o valor de um carácter e, se este for uma letra minúscula, retorna o valor da maiúscula correspondente; nas outras situações retorna o valor recebido, sem modificação.

```
int toupper(int c);
```

Se precisar de verificar se um carácter corresponde a uma letra minúscula, utilize a função islower, que recebe o valor de um carácter, retornando um valor diferente de zero se este corresponder a uma minúscula.

```
int islower(int c);
```

b) [4] Considere as seguintes definições:

```
struct StrToConvert { unsigned int nconverted; char * str; };
unsigned int ConvertAllStrs(struct StrToConvert toConv[],
unsigned int nconvs,
```

```
char * (*strconvf)(char *, unsigned int *));
```

Implemente, em assembly IA-32, a função ConvertAllStrs, que aplica a função de conversão apontada por strconvf às strings apontadas pelos campos str de cada um dos nconvs elementos do array toConv, retornando o número total de caracteres convertidos.

A função de conversão apontada por strconvf recebe, como primeiro argumento o endereço da string a converter e, como segundo argumento, o ponteiro para a localização onde deixar o número de caracteres convertidos. Retorna o endereço da sequência convertida.

3. (2 valores)

Para a concepção de uma cache, considerando uma determinada dimensão de RAM de dados, apresente as vantagens e inconvenientes de optar por mapeamento directo ou 2-way set associative.

4. (2 valores)

Numa cache do tipo 4-way set associative, cada valor de SET (INDEX) selecciona um campo de TAG ou quatro campos de TAG em simultâneo? Justifique.

8. (5 valores)

Escreva, em assembly IA-32, o código da função

```
char * getStringAddress(int count, const char * match, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de ponteiros para strings, com tipo char *, e compara cada uma das strings com a apontada por match, terminando quando tiver encontrado count strings idênticas a match e retornando o endereço da última delas.

Se não houver strings idênticas em número suficiente, retorna NULL. A lista de parâmetros variável é terminada por um ponteiro com o valor NULL.

Para fazer as comparações, deve utilizar a função de biblioteca

```
int strcmp( const char * str1, const char * str2 );
```

que retorna zero se as strings apontadas por str1 e str2 forem idênticas.

4. (3,5 valores)

Considere um processador com cache interna L1 e L2. Admita que a cache L2 é 4-way set associative, com 512 Kbyte de dados em linhas de 32 bytes.

- a) (1) Qual é a dimensão, em bits, da memória de Tag? Apresente os cálculos.
- b) (1) Que vantagens e inconvenientes teria a modificação para uma organização 8-way?
- c) (1) Que vantagens e inconvenientes teria a modificação da dimensão da linha para 64 bytes?
- d) (0,5) Admitindo a possibilidade de dispor de maior área de circuito integrado para memória de cache, explique os critérios a seguir para utilizar essa memória em Data ou em Tag.

8. (4 valores)

Considere as seguintes definições:

```
typedef struct {  
    void * context;  
    int (*func)(void * ctx, int n, ...);  
} Function;  
  
int TestAll(Function * f[], int nfuncs, int nwords, ...);
```

Implemente, em assembly IA-32, a função TestAll, que, para cada um dos nfuncs elementos de f, chama a função indicada pelo respectivo campo func, passando como argumentos o valor do campo context, o valor de nwords e as nwords words recebidas por TestAll na lista de argumentos variável. A função TestAll retorna a soma dos valores retornados pelas várias chamadas.

4. [4 valores]

Nos processadores Intel baseados na microarquitetura Core, há uma versão em que a cache de nível 2 é associativa de 8 vias com 2 Mbyte dados em linhas de 64 byte.

- a) [2] Considerando que o endereçamento destes processadores é a 36 bit, determine o número de linhas e os bits de endereço que definem os campos BYTE OFFSET, SET (INDEX) e TAG.
- b) [1] Noutra versão, a RAM de dados da cache tem 4 Mbyte. Indique as opções possíveis relativamente aos parâmetros da cache a alterar para ter esta dimensão.
- c) [1] Apresente as razões que no seu entender motivaram o fabricante a optar por duplicar o número de vias.

Admita a definição dos campos do endereço, para acesso a uma cache L2, como é mostrado na figura

A35 A18 A17 A6 A5 A0
TAG SET (INDEX) BYTE OFFSET

- a) [2] Para a cache com estas definições, qual é a dimensão total mínima da memória de dados? Caracterize este sistema de cache, relativamente a número de vias, dimensão de cada via e dimensão da linha.
- b) [2] Mantendo os campos idênticos, duplicou-se o número de vias. Caracterize o novo sistema de cache.
- c) [2] Enumere as vantagens e desvantagens da solução apresentada na alínea anterior.
- d) [2] Indique a relação que existe entre: dimensão da TAG e a dimensão da memória de dados; dimensão do SET (INDEX) e a dimensão da memória de dados.

3. (x valores)

Admita a existência de um sistema de cache L2 2-way set associative, com as seguintes características:

TAG: 15 bits
INDEX(SET): 12 bits
BYTE OFFSET: 5 bits

- a) [x] Determine a dimensão da memória de dados.
- b) [x] Determine a capacidade (física) de endereçamento do sistema.
- c) [x] Mantendo os campos TAG, INDEX, BYTE OFFSET, retirou-se uma way (via). Determine a nova dimensão da memória de dados.
- d) [x] Mantendo a memória de dados com a mesma dimensão determinada na alínea a), retirou-se uma way (via). Determine, em bits, a dimensão dos campos TAG, INDEX e BYTE OFFSET.
- e) [x] Indique, justificando, relativamente aos três sistemas de cache, o inicial, o determinado na alínea c) e o determinado na alínea d), por qual optava.

4. (1 valor)

No contexto dos sistemas de cache n-way set associative, indique, justificando, se é verdadeira ou falsa a afirmação: "para carregar um novo conteúdo, pode ocorrer a necessidade de substituir uma linha preenchida, ainda que outras linhas estejam vazias"

5. (3 valores)

Considerando uma cache de nível 2 com 256 kbyte de dados em linhas de 64 bytes, com organização 4-way set-associative, num computador com endereçamento a 32 bits, determine:

- a) [1] Os bits de endereço que formam os campos byte-offset, set e tag;
- b) [1] A dimensão total, em bits, da memória de gestão que inclui os campos de Tag, Valid e Modified (admitindo que a política de escrita é write-back);
- c) [1] As alterações da memória de gestão se a organização fosse alterada para 8 vias, mantendo a dimensão total da memória de dados e o comprimento das linhas.

1. (5 valores)

Escreva, em assembly IA-32, o código da função

```
char *strNesimaStr(const char *str1, const char *str2, int nesima);
```

que retorna o apontador para a n-ésima ocorrência do conteúdo da string str2 na string str1. Se não existir o número de ocorrências pretendido, retorna o valor NULL.

Para localizar as ocorrências da string deve utilizar as funções de biblioteca

```
char * strstr(const char *cs, const char *ct);
```

que retorna o apontador para primeira ocorrência da string ct na string cs, ou NULL se não existir;

```
int strlen(const char * cs);
```

que retorna a dimensão da string cs.

2. (5 valores)

Dada a estrutura

```
struct FuncDsc {  
    void (*func)(char, int);  
    int i;  
    char c;  
};
```

Escreva, em assembly IA-32, o código da função

```
void FuncCaller(struct FuncDsc *f);
```

que recebe um apontador para a estrutura FuncDsc e realiza a chamada à função func, com os parâmetros i e c.

4. [4 valores]

Comente a seguinte afirmação: "A utilização da política de actualização write-through em caches permite garantir a consistência de dados no sistema de uma forma simples. Contudo, esta simplicidade traduz-se numa relativa degradação no desempenho do sistema."

5. [6 valores]

Considere, num sistema com endereçamento a 32-bits, uma cache L2 do tipo associativo a 4-vias com 2 Mbyte de dados em linhas de 32 bytes.

a) [2] Determine o número de bits no barramento de endereço usados para INDEX (SET) e OFFSET.

b) [2] Determine a capacidade, em bits, da memória de TAG.

c) [2] Indique, justificando, a melhor configuração para a cache L1 deste sistema de forma a maximizar a sua eficiência:

i. uma cache associativa de 2-vias com 8 Kbyte de dados em linhas de 32 bytes;

ii. uma cache associativa de 4-vias com 8 Kbyte de dados em linhas de 16 bytes;

Dada a estrutura

```
struct UserDsc { long id; short number; char status; void * class; };
```

Escreva, em assembly IA-32, o código da função

```
void backup(struct UserDsc *src, struct UserDsc *dst, int nelem);
```

que copia o conteúdo do array apontado por src, com nelem elementos, para o array apontado por dst. A função devolve o número de bytes transferidos.

7. [6 valores]

Escreva, em assembly IA-32, o código da função

```
int getStrAdr(char *dst[], int max, char * sample, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de endereços de string, terminado por NULL, e retorna o número de strings com início idêntico a sample; preenche o array dst com os endereços dessas strings, limitando pelo valor de max o número de elementos preenchidos.

Para identificar as strings deve utilizar as funções de biblioteca

```
int strncmp(const char *str1, const char *str2, int count);
```

que compara os primeiros count caracteres de str1 e str2, retornando 0 se forem iguais;

```
int strlen(const char * str);
```

que retorna a dimensão da string str.

5. [1 valor]

Considere as duas versões seguintes de um troço de código em linguagem C

Versão A Versão B

```
int i, j, s=0;  
for(i=0; i<MAX_L; ++i)  
for(j=0; j<MAX_C; ++j)  
s += array[j][i];  
int i, j, s=0;  
for(j=0; j<MAX_C; ++j)  
for(i=0; i<MAX_L; ++i)  
s += array[j][i];
```

Admitindo que o código é executado num computador com cache, indique, justificando, se alguma das versões é preferível.

6. [4 valores]

Escreva, em assembly IA-32, o código da função

```
int copyAbove(int *dst[], unsigned int limit, int n, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de n valores inteiros, copia para o array dst os que tiverem valor absoluto superior a limit e retorna o número de elementos copiados.

Para determinar o valor absoluto dos parâmetros deve utilizar a função de biblioteca int abs(int v);

que retorna o valor absoluto do parâmetro v.

7. [1 valor]

Considere que foi executado o assembler as, tendo como input as linhas seguintes:

```
.intel_syntax noprefix
```

```
mov [ebx], 0x35
```

```
mov eax, [ecx]
```

```
mov [edi], [esi]
```

e produziu estas mensagens:

```
2: Error: ambiguous operand size for 'mov'
```

```
4: Error: too many memory references for 'mov'
```

Explique o significado, a causa e como se pode resolver cada um dos erros indicados.

6. [x valores]

Escreva, em assembly da arquitectura IA-32, a função void copyPlus(int dst[], ...); que preenche o array dst com os elementos, do tipo int, positivos da lista de parâmetros variável que termina com o valor -1.

4. [6 valores]

Considere um sistema com um processador "Pentium M" com endereçamento a 36 bit, dispondo de caches L1 com 32 kbyte para código e 32 kbyte para dados, ambas do tipo 4-way set associative, e de cache L2, unificada, do tipo 8-way set associative com 1 Mbyte em linhas de 64 byte.

- [2] Indique, justificando, a dimensão das linhas das caches L1.
 - [2] Determine, em bits, a capacidade da memória de TAG da cache de dados L1.
 - [2] Sabendo-se que um maior grau de associatividade permite obter hit-rates mais elevados, justifique a organização adoptada para a cache de dados L1 comparativamente com a organização da cache L2.
-

5. [4 valores]

Considerando um controlador de cache com funcionamento em modo write-back e uma política de substituição random, descreva as operações desencadeadas por um miss de escrita, admitindo os seguintes cenários para a linha a substituir: i) está inválida; ii) está válida não modificada; iii) está válida e modificada.

1. (2 valores)

Considerando as definições

```
typedef struct {int a, int b} sX;  
typedef struct {int a, int b, int c} sY;  
sX fX(int x);  
sY fY(int y);
```

Indique, justificando, quantos parâmetros recebe em stack cada uma destas funções.

2. (3 valores)

Escreva, em assembly da arquitectura IA-32, o código da função

```
int charinrange(char c, char first, char last);
```

que devolve 1 se valor de c está compreendido no intervalo de first a last, inclusive; caso contrário, devolve 0.

(6 valores)

Considerando a definição

```
typedef struct {int lower, int upper} charcount;
```

escreva, em assembly da arquitectura IA-32, o código da função

```
charcount charstat(...);
```

que recebe na lista de parâmetros variável, terminada por NULL, os endereços de várias strings e devolve, nos campos lower e upper, respectivamente, o número total de minúsculas e maiúsculas existentes nessas strings. Para determinar a gama a que pertence cada carácter, deve utilizar a função charinrange.

[9 valores]

Considere um sistema com endereçamento a 32 bits e apenas um nível de cache, do tipo 4-way set associative, com 8 kB em linhas de 16 byte e uma política de escrita do tipo write-through.

- [2] Determine, em bits, a dimensão dos campos OFFSET, SET e TAG.
- [2] Determine, em bits, a capacidade da memória de TAG.
- [2] Indique as principais vantagens e inconvenientes da política de escrita adoptada.
- [3] Com vista a aumentar o desempenho do sistema pretende-se substituir esta cache.

Discuta as vantagens e desvantagens das seguintes soluções:

Cache do tipo direct-mapped com capacidade 16 kB;

Cache do tipo 8-way set associative com capacidade 8 kB.

4. [2 valores]

Considere que foi executado o assembler as, tendo como input as linhas seguintes:

```
.intel_syntax noprefix  
add eax, byte ptr [ebx]  
add ebx, 1
```

e produziu esta mensagem:

```
2: Warning: using '%al' instead of '%eax' due to 'b' suffix
```

Explique o significado, a causa e como se pode resolver o problema indicado, considerando que se pretende efectivamente adicionar um byte ao registo EAX.

5. [3 valores]

Escreva, em assembly da arquitectura IA-32, o código da função int accumulate(short val, int sum); que devolve a soma de val com sum.

6. [7 valores]

Escreva, em assembly da arquitectura IA-32, o código da função int zeroMedian(short a[], int n);

que determina a média aritmética dos n elementos do array e subtrai esse valor a cada um dos elementos, de modo que, no final, a média dos elementos seja 0. A função retorna a média aritmética calculada.

Como auxiliar para o cálculo da média, deve utilizar a função accumulate.

4. [4 valores]

Considere um processador com endereçamento a 36 bit, dispondo de cache interna com dois níveis. A cache L2 é unificada, do tipo 8-way set associative, com 1 Mbyte de dados em linhas de 64 bytes, com políticas de escrita write-back e de substituição random.

- [1] Determine a dimensão total, em bits, da memória necessária para esta cache, incluindo dados, tags, bits V (valid) e M (modified). Apresente todos os cálculos, devidamente identificados.

- b) [1] Determine, em percentagem, o acréscimo de memória necessário para duplicar o número de vias. Apresente todos os cálculos, devidamente identificados.
- c) [1] Enumere, justificando, os critérios a considerar para definir o número de vias, no projecto da cache, e indique a sua relevância relativa.
- d) [1] Discuta as vantagens e inconvenientes de usar a política de substituição indicada.

5. [1 valor]

Justifique o facto de, nas caches do tipo n-way set associative , os bits de endereço usados para tag serem os de maior peso.

6. [1,5 valores]

Relativamente à preservação de registos nas chamadas a funções:

- a) [0,5] Indique, justificando, os registos cuja preservação é indispensável;
- b) [0,5] Indique os registos, além dos anteriores, que são preservados nas convenções adoptadas pelas ferramentas GNU, descrevendo as vantagens de fazer essa preservação;
- c) [0,5] Justifique o facto de os restantes registos não serem preservados.

7. [2 valores]

Escreva, em assembly da arquitectura IA-32, o código da função

```
int limitedCopy(char * d, char * s, int dSize);
```

que copia a string s para o endereço d , limitando a dSize o número de caracteres copiados, incluindo o terminador de string . Devolve 1 se copiou a totalidade da string ou 0, caso contrário.

8. [3 valores]

Escreva, em assembly da arquitectura IA-32, o código da função

```
int limitedConcat(char * dst, int dstSize, ...);
```

que recebe na lista de parâmetros variável, terminada pelo pointer NULL, um conjunto de strings e copia o seu conteúdo, pela ordem em que estão na lista, para o endereço dst , limitando a dstSize o número de caracteres copiados, incluindo o terminador de string . Devolve 1 se copiou a totalidade das strings ou 0, caso contrário.

Para executar o preenchimento de dst deve usar a função limitedCopy() , definida na pergunta anterior.

4. [3 valores]

Considere, associado a um processador com endereço a 36 bits, um sistema de cache do tipo 8-way set associative cuja RAM de dados tem a dimensão total de 1 Mbyte, em linhas de 32 bytes.

- a) [1] Determine quais os bits de endereço utilizados para byte offset, set (index) e tag. Justifique a posição relativa de cada um destes grupos de bits.
- b) [1] Determine a quantidade total, em bits, de memória para armazenamento de tags.
- c) [1] Descreva as vantagens e inconvenientes de modificar esta cache, indicando justificadamente, quais dos valores calculados nas alíneas anteriores são modificados, relativamente a duas opções:
- i) reduzir o número de vias para 4;
- ii) aumentar a dimensão da linha para 64 bytes.

6. [3 valores]

Escreva, em assembly da arquitectura IA-32, a função int proper(char * s);

que modifica a string recebida por parâmetro, convertendo para maiúscula o primeiro carácter de cada palavra e para minúscula os restantes. Considere as palavras separadas pelo carácter espaço. A função devolve o número de palavras existentes. Para fazer as conversões deve utilizar as funções de biblioteca

```
int toupper(int c); // converte para maiúscula
```

```
int tolower(int c); // converte para minúscula
```

7. [2 valores]

Escreva, em assembly da arquitectura IA-32, a função

```
int properList(int count, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de count endereços de string e executa sobre cada uma delas a função proper definida na questão anterior. A função properList devolve o número total de palavras existentes nas strings processadas.