

Entregue um documento com as respostas às questões apresentadas. Contacte o docente se tiver dúvidas. Encoraja-se a discussão de problemas e soluções com colegas de outros grupos, mas recorda-se que a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

1. Dado um sistema de *cache* de 16KiB, 4-way *set associative*, com linhas de 32 *bytes*, indique (apresentando os respectivos cálculos, quando se aplique):
 - a) Caracterize a divisão dos endereços em *tag*, *index* e *offset*, sabendo que a *cache* é utilizada num sistema com um *address bus* de 36 *bits*.
 - b) Considere `sizeof(int) == 4`, a variável global `int dat[N]` e a função `cyclic_read`, que consiste num ciclo infinito de leitura de todas as posições do *array* `dat` e que utiliza apenas os registos do processador para manter as variáveis auxiliares de que necessita. Qual o menor valor de *N* que garante que duas entradas do *array* `dat` pertencem ao mesmo *set* da *cache*? E qual o menor valor de *N* que garante a ocorrência de *cache misses* da segunda passagem no *array* em diante, mesmo que existam *caches* separadas para código e dados?
2. Considere as implementações das funções `xstrcat` e `astrcat`, realizadas nas séries de exercícios anteriores, bem como as implementações `xlstrcat` e `x2strcat` apresentadas a seguir e a implementação de `strcat` da biblioteca *standard*, disponibilizada pelas ferramentas de desenvolvimento que utiliza. Meça o tempo de execução destas cinco funções para *strings* *s1* e *s2* com dimensões iguais, compreendidas entre 16B e 8KiB em passos de 16 *bytes*, apresentando os resultados em tempo absoluto de forma gráfica. Comente os aspectos relevantes dos resultados.

Data limite de entrega: 10 de Maio de 2010

Bom trabalho!

```
unsigned int xstrlen(const char *s) {
    const char * p=s;
    while (*(p++));
    return p-s-1;
}

char * xlstrcat(char * s1, const char * s2) {
    char *s = s1 + xstrlen(s1);
    while( xstrlen(s2) ) *(s++) = *(s2++);
    return s1;
}

char * x2strcat(char *s1, const char *s2) {
    char *s = s1;
    if (*s) x2strcat(s+1,s2);
    else while( (*(s++)=*(s2++)) );
    return s1;
}
```