Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Programação em Sistemas Computacionais

Verão de 2015/2016

Série de Exercícios #1 - Individual

Realize os exercícios seguintes usando a linguagem C. Não se esqueça de testar devidamente o código desenvolvido, bem como de o apresentar de forma cuidada, apropriadamente indentado e comentado. Assegure-se de que o compilador não emite qualquer aviso sobre o seu código, mesmo com a opção `-Wall` activa. Contacte o docente se tiver dúvidas. Não é necessário relatório. Encoraja-se a discussão de problemas e soluções com outros colegas, mas a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

1. Implemente a função `_strstr`, que será a sua versão da função `strstr` da biblioteca *standard* da linguagem C, sem recorrer a outras funções dessa biblioteca (use o comando `man strstr` para obter informação sobre esta função).

2. Considere vectores de *bits* representados sobre *arrays* de inteiros e que `INT_BIT` é o número de *bits* de um inteiro. No valor de índice `n` de um *array* de inteiros, o *bit* de menor peso corresponde ao índice `n * INT_BIT` do vector de *bits* e o *bit* de maior peso corresponde ao índice `(n + 1) * INT_BIT - 1 do vector de bits`. A função `vgetbits` retorna o valor dos *bits* entre as posições `idx` e `idx + len - 1` do vector de *bits* representado por *data*. A função `vsetbits` escreve os *len bits* de menor peso de *val* nas posições entre `idx` e `idx + len - 1` do vector de *bits* representado por *data*. Em ambas as funções, `len` nunca é maior do que `INT_BIT`. Ex.: para `data = { 0xABCD1234, 0xFFFFFFEC }`, a chamada a `vgetbits(data, 29, 8)` retorna `0x00000065`. Defina `INT_BIT` e escreva as funções `vgetbits` e `vsetbits`.

   ```
   unsigned vgetbits(unsigned data[], unsigned idx, unsigned len);
   void vsetbits(unsigned data[], unsigned idx, unsigned len, unsigned val);
   ```

3. Apresente a função `is_short`, que retorna *true* se e só se o valor recebido como argumento puder ser representado por um `short` sem perda de informação. Na implementação interna só podem ser utilizadas operações aritméticas e lógicas sobre inteiros. Qualquer operação de vírgula flutuante invalida o exercício.

   ```
   bool is_short(float f);
   ```

4. Considere a definição apresentada do tipo `struct hmstime` e desenvolva a função `sumtimes`, que soma os tempos dos `ntimes` elementos do *array times* e deixa o resultado, devidamente transformado, na instância de `struct hmstime` referida por `res`.

   ```
   struct hmstime { unsigned short hours; unsigned char minutes; unsigned char seconds; };
   void sumtimes(struct hmstime * res, const struct timeval times[], size_t ntimes);
   ```

   Nota: use o comando `man gettimeofday` para obter a definição da estrutura `timeval`.

Data limite de entrega: 3 de Abril de 2016

ISEL, 16 de Março de 2016