Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Programação em Sistemas Computacionais

Verão de 2017/2018

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 `_strrchr`, que será a sua versão da função `strrchr` da biblioteca *standard* da linguagem C, sem recorrer a outras funções dessa biblioteca.

2. Considere vectores de *bits* representados sobre *arrays* de *unsigned long* e que `ULONG_BIT` é o número de *bits* do tipo *unsigned long*. No índice `n` de um *array* de `unsigned long`, o *bit* de menor peso corresponde ao índice `n * ULONG_BIT` do vector de *bits*, enquanto que o *bit* de maior peso corresponde ao índice `(n + 1) * ULONG_BIT - 1`. A função `getbits` retorna o valor dos *bits* entre as posições `idx` e `idx + len - 1` do vector de *bits* representado por `bits`. A função `setbits` escreve os `len` *bits* de menor peso de `val` nas posições entre `idx` e `idx + len - 1` do vector de *bits* representado por `bits`. Em ambas as funções, considere que `len` nunca é maior do que `ULONG_BIT`. Ex.: para `bits = { 0x98761111FCDFEC80, 0xABCDEF001234567A }`, a chamada a `getbits(bits, 56, 20)` retorna `0x0000000000067A98`. Usando a linguagem C, defina `ULONG_BIT` e escreva as funções `getbits` e `setbits`.

   ```
   unsigned long getbits(unsigned long bits[], unsigned idx, unsigned len);
   void setbits(unsigned long bits[], unsigned idx, unsigned len, unsigned long val);
   ```

3. Desenvolva a função `ftoi`, que retorna a parte inteira de `val`. Se a parte inteira de `val` exceder `INT_MAX`, a função retorna `INT_MAX`. Se a parte inteira de `val` for menor do que `INT_MIN`, a função retorna `INT_MIN`. Na implementação interna só podem ser utilizadas operações aritméticas e lógicas sobre inteiros. Qualquer operação de vírgula flutuante invalida o exercício.

   ```
   int ftoi(float val);
   ```

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`.

5. Escreva o programa utilitário *within*, que copia para o *standard output* as linhas do *standard input* que contenham pelo menos uma das *strings* passadas como argumento ao programa.

Data limite de entrega: 1 de Abril de 2018

ISEL, 5 de Março de 2018