

# PUC-Rio – Software Básico – INF1018

## Prova 1 – Turma 3WB – 06/10/2022

---

1) (3,0 pontos) Considere o programa C a seguir:

```
#include <stdio.h>

struct S {
    char a;
    int b;
    int *c;
    char d;
};

void dump(void *p, int n);

int main(void) {
    struct S s;
    s.a = 100;
    s.b = 'L';
    s.c = &s.b;
    s.d = 'D';
    dump(&s, sizeof s);
    return 0;
}

void dump(void *p, int n) {
    unsigned char *p1 = (unsigned char *)p;
    while (n--) {
        printf("%p - %02x\n", p1, *p1);
        p1++;
    }
}
```

Suponha que a máquina de execução seja little-endian com as convenções de alinhamento do Linux no IA-64 vistas em sala. Além disto, considere que o endereço de `s` é `0x7ffe8ef78240` e que o valor do caractere 'A' na tabela ASCII é `0x41` ou `65`, responda:

- Faça uma representação pictográfica da estrutura `s` na memória de um computador, informando o endereço e o nome de cada campo. Informe onde foi necessário acrescentar bytes usando a expressão `PADDING`.
- Descreva o que este programa irá imprimir quando executado, justificando os valores exibidos mostrando as contas e outras informações usadas para chegar ao resultado.

2) (2,0 pontos) Implemente uma função em C que receba três bytes sem sinal e retorne um inteiro sem sinal extraindo os bits mais significativos como mostrado na figura a seguir.

Valores recebidos	Valor retornado
$1110b_{15}b_{14}b_{13}b_{12},$ $10b_{11}b_{10}b_9b_8b_7b_6,$ $10b_5b_4b_3b_2b_1b_0$	$b_{15}b_{14}b_{13}b_{12}b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0$
Exemplos (valor em binário e o equivalente em decimal)	
$11100010$ (226), $10010100$ (148), $10011011$ (155)	$000000000010010100011011$ (619)
$11100110$ (230), $10100001$ (161), $10010000$ (144)	$000000000110100001010000$ (1744)
$11100010$ (226), $10101010$ (170), $10111101$ (189)	$00000000010101010111101$ (861)

Use o seguinte protótipo.

```
unsigned int extrai(unsigned char b0, unsigned char b1, unsigned
char b2);
```

3) (5,0 pontos) Traduza a função boo abaixo para assembly IA-64 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/Linux. Traduza o mais diretamente possível o código de C para assembly.

**Importante: comente o seu código!**

```
int funcao(int item, int peso);

int boo(int *nums, int tam) {
    int valor;
    valor = 0;
    for (int i = 0; i < tam; i++) {
        if (nums[i] > valor) {
            valor = funcao(nums[i], valor);
        }
    }
    return valor;
}
```

Boa prova!