Prova Final de Software Básico — 2008.2

Duração: 1 hora e 45 minutos

27 de novembro de 2008

Atenção: Seja preciso e objetivo em suas respostas. Comente suas respostas: clareza conta ponto. Não esqueça de colocar o seu nome e número de matrícula na folha de respostas. **Boa sorte!**

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

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

struct X {
   char c;
   float f;
   short s;
} a[] = {{-1 << 3, -10.0, -10},
         {~('h' - 'b'), 11.5, 0x78 & 65},
         {~((~0) << 5), -0.75, -1000}};

int main (void) {
   dump (a, sizeof(a));
   return 0;
}
```

   Diga o que este programa irá imprimir quando executado, justificando os valores exibidos e mostrando as suas contas. Suponha que a máquina de execução é *little endian* e que o vetor `a` começa no endereço 0x2A000000. Escreva 'XX' nos bytes que contiverem lixo.

2. (2 pontos) Traduza a função a seguir para Assembly da família de processadores Intel Pentium (IA-32), utilizando as regras usuais do C/Linux para passagem de parâmetros, de valor de retorno, e de salvamento de registradores na pilha. **Atenção:** Comente o seu código.

```
double func (float x, int d) {
  double res = 1.0;
  if (d >= 10) {
    while (d) {
      res = res * (double)x;
      d--;
    }
  }
  return res;
}
```

3. (3 pontos) Traduza a função a seguir para Assembly da família de processadores Intel Pentium (IA-32), utilizando as regras usuais do C/Linux para passagem de parâmetros, de valor de retorno, e de salvamento de registradores na pilha. **Atenção:** Comente o seu código.

```
struct arv {
  double x;
  int d;
  struct arv *fe;
  struct arv *fd;
};

double somaNos (struct arv *a, double *d) {
  double r;
  if (a == NULL) r = d[0];
  else r = d[a->d] + somaNos(a->fe, d) + somaNos(a->fd, d);
  return r;
}
```

4. (2 pontos) Considere a declaração abaixo:

```
int classificafloat (float f);
```

Escreva a função `classificafloat` em assembler. Ela deve receber um número float `f` e retornar um código de classificação do número, de acordo com o critério abaixo:

"classe" de `f`	retorno
float normalizado	0
float denormalizado	1
infinito positivo	2
infinito negativo	3
Not-a-Number	4