UFPR – Bacharelado em Ciência da Computação
CI064 – Software Básico— 2014-2

Primeira Prova

Para facilitar a correção indique os registradores que não são usados na convenção de chamada de funções como `re`, `rn`, *etc. Use uma folha inteira para escrever cada programa em assembly.*

1. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [10 pontos]

```
int power(int n,int exp) {
   if (exp > 1)
      return (n * power(n, exp-1));
   else
      return (n);
}

void main (void) {
   int x,y,z;

   y = 5; z = 3;
   x = power(y,z);
   printInt(x);
}
```

2. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [10 pontos]

```
int a, i;
int x[2048], y[256];
...
i=0;
a=0;
while (i < 1024) {
   a = a + x[i] + y[ (x[i] % 256) ]; // MOD
   i = i + 1;
}
```

3. Esta questão examina a programação no ambiente Unix e tem três itens. Por 'programa' entenda-se uma sequência de comandos e executáveis, ou o conteúdo de um *script*.

(a) Escreva um programa que lista os *usernames* de todos os alunos do BCC cujos diretórios `$HOME` não estejam devidamente protegidos contra acessos indevidos. [5 pontos]

(b) Escreva um programa que lista todos os arquivos em seu diretório `$HOME` que possuam permissão de escrita para outros que não você próprio. [5 pontos]

(c) Escreva um programa que lista os *usernames* de todos os alunos do BCC em cujo *username* ocorra a letra `w` (dábliu, ou doblevê). A saída deve ser em ordem alfabética. [5 pontos]

Segunda Prova

Convenção:
(i) a definição de um símbolo é denotada nos diagramas por um *label* como `simb:`
(ii) uma referência a uma função é denotada pelos parênteses como `fun()`
(iii) uma referência a uma variável é denotada pelo seu nome como `var` .

4. Esta questão tem dois itens. (a) Descreva o processo de ligação com bibliotecas compartilhadas estáticas (BCEs). Sua resposta deve conter o algoritmo e as estruturas de dados necessárias. Se o processo de ligação envolve a geração de código *assembly* para alguma finalidade, escreva um exemplo deste código. (b) Usando o código abaixo e as duas bibliotecas, execute o algoritmo do item (a) e preencha as estruturas de dados com os valores indicados no código. [20 pontos]

No código C, o endereço nos comentários é o endereço da instrução que invoca a função. Para simplificar sua resposta, variáveis atribuídas são referenciadas em endereço que é 4 bytes maior que o da instrução `jal`, enquanto que variáveis lidas são referenciadas num endereço que é 4 bytes menor do que o da instrução `jal`: em `main()`, `alpha` é referenciada em 0x0404 e `x` em 0x03fc.

```
extern int foo(int);
extern int bar(int);
extern int cat(int);
extern int rat(int);
int fun(int);
extern int alfa;
extern int beta;
extern int gama;

int main(int argc, char **argv) {
  int x,y,z;
  ...
  alfa = fun(x); // 0x0400
  ...
  beta = foo(z); // 0x1000
  ...
  y = bar(alfa); // 0x2400
  ...
  return(z);     // 0x24fc
}

int fun(int) {
   int x,y,z;
   ...
   gama = cat(x); // 0x3000
   ...
   beta = rat(y); // 0x3800
   ...
   z = foo( bar(alfa) ); //0x4000
   ...
   return(y);     // 0x4400
}
```

```
// libfoobar
.org  0x1000 # ender = 0x8.1000
.text
foo: ...
...
.org  0x2000 # ender = 0x8.2000
.text
bar: ...
...
.org  0x3000 # ender = 0x8.3000
.data
alfa: .space 4,0 # int
beta: .space 4,0 # int
.end libfoobar
//------------------------------

// libcatrat
.org  0x2000 # ender = 0x9.2000
.text
cat: ...
...
.org  0x3000 # ender = 0x9.3000
.text
rat: ...
...
.org  0x4000 # ender = 0x9.4000
.data
gama: .space 4,0 # int
.end libcatrat
//------------------------------
```

5. Esta questão tem três itens e sua resposta NÃO pode conter as diretivas da questão anterior. (a) Dê dois exemplos de diretivas do montador que *não* causam a inclusão de bits adicionais no arquivo objeto e explique suas funções; (b) dê dois exemplos de diretivas do montador que produzem saída no arquivo objeto e explique suas funções; (c) qual a função da diretiva `.align` ? Esta diretiva não pode estar incluída nas suas respostas anteriores. [5 pontos]

6. Desenhe um diagrama de um arquivo objeto no formato ELF para um executável (que não é uma biblioteca) e explique para que serve cada componente do arquivo. [5 pontos]

7. Considere o programa de multiplicação de matrizes mostrado abaixo. Suponha que as matrizes contém 1024x1024 elementos, cada elemento um **`float`**. O programa é executado num único processador com páginas de 4Kbytes. Descreva o comportamento do sistema de memória virtual durante a execução deste programa, supondo que o programa foi inicializado há muito tempo (segundos) e que as matrizes não foram referenciadas desde a inicialização. [5 pontos]

```
int i,j,k;
float sum, A[1024][1024], B[1024][1024], C[1024][1024];

for (i = 0; i < 1024; i++) {
   for (j = 0; j < 1024; j++) {
      for (sum = 0, k = 0; k < 1024; k++)
         sum += A[i][k] * B[k][j];
      C[i][j] = sum;
   }
}
```

Exame Final

8. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS[1]. [35 pontos]

```
#define SIZE 1024

typedef struct x {
   int a;
   int b;
   int c;
   short x;
   short y;
} xType;

xType V[SIZE];
xType Z[SIZE];

void reduz(int lim, xType *v, xType *z, int pot) {
   int i=0;
   while (i < lim) {
      v[i].a = z[i].b + z[i].c;
      v[i].x = z[i].x <<pot;
      i = i + 1;
   }
}
   ...
   reduz(SIZE/4, V,  Z, 4);
   ...
```

[1]Para facilitar a correção indique os registradores que não são usados na convenção de chamada de funções como rp, rq, etc. Use uma folha nova/inteira para responder esta questão.

9. Descreva a organização do *driver* da interface serial do trabalho desta disciplina. Sua descrição deve conter as estuturas de dados do *driver* e uma descrição em pseudo-C do seu comportamento. [30 pontos]

10. Esta questão tem três itens. (a) Descreva o processo de construção/compilação de uma biblioteca compartilhada dinâmica (BCD) e indique quais são as estruturas de dados necessárias e como elas são usadas na construção da biblioteca. (b) Descreva o processo de ligação com BCDs. Se o processo de ligação envolve a geração de código *assembly* para alguma finalidade, escreva um exemplo deste código. (c) Usando o código abaixo e as duas bibliotecas, execute o algoritmo do item (a) e preencha as estruturas de dados com os valores indicados no código. [35 pontos]

No código C, o endereço nos comentários é o endereço da instrução que invoca a função. Para simplificar sua resposta, variáveis atribuídas são referenciadas em endereço que é 4 bytes maior que o da instrução `jal`, enquanto que variáveis lidas são referenciadas num endereço que é 4 bytes menor do que o da instrução `jal`: em `main()`, `alpha` é referenciada em 0x0404 e `x` em 0x03fc.

```
extern int foo(int);
extern int bar(int);
extern int cat(int);
extern int rat(int);
int fun(int);
extern int alfa;
extern int beta;
extern int gama;

int main(int argc, char **argv) {
  int x,y,z;
  ...
  alfa = fun(x); // 0x0400
  ...
  beta = foo(z); // 0x1000
  ...
  y = bar(alfa); // 0x2400
  ...
  return(z);     // 0x24fc
}

int fun(int) {
   int x,y,z;
   ...
   gama = cat(x); // 0x3000
   ...
   beta = rat(y); // 0x3800
   ...
   z = foo( bar(alfa) ); //0x4000
   ...
   return(y);     // 0x4400
}
```

```
// libfoobar
.org  0x1000 # ender = 0x8.1000
.text
foo: ...
...
.org  0x2000 # ender = 0x8.2000
.text
bar: ...
...
.org  0x3000 # ender = 0x8.3000
.data
alfa: .space 4,0 # int
beta: .space 4,0 # int
.end libfoobar
//------------------------------

// libcatrat
.org  0x2000 # ender = 0x9.2000
.text
cat: ...
...
.org  0x3000 # ender = 0x9.3000
.text
rat: ...
...
.org  0x4000 # ender = 0x9.4000
.data
gama: .space 4,0 # int
.end libcatrat
//------------------------------
```