

**Professores:** Fernando V. Paulovich (paulovic at icmc.usp.br)  
Moacir Ponti Jr (moacir at icmc.usp.br)  
**Aluno PAE:** Felipe S. L. G. Duarte (fgduarte at icmc.usp.br)  
**Monitor:** Cassiano K. Casagrande (cassianokleinert at gmail.com)

## Trabalho 01: Júpiter Not Web

### 1 Prazos e Especificações:

O trabalho descrito a seguir é individual e não será tolerado qualquer tipo de plágio ou cópia em partes ou totalidade do código. Caso seja detectado alguma irregularidade, os envolvidos serão chamados para conversar com o professor responsável pela disciplina.

A entrega deverá ser feita única e exclusivamente por meio do Sistema de Submissão de Programas (SSP) no endereço eletrônico <http://ssp.icmc.usp.br> até o dia **31 de março de 2013 as 23 horas e 59 minutos**. Sejam responsáveis com o prazo final para entrega, o SSP está programado para não aceitar submissões após este horário e não será aceito entrega fora do sistema. Não serão aceitas submissões por email.

Leia a descrição do trabalho com atenção e várias vezes, anotando os pontos principais e as possíveis formas de resolver o problema. Comece a trabalhar o quanto antes para que não ficar dúvidas e você consiga entregar o trabalho a tempo.

O trabalho deverá ser submetido em formato zip/Makefile contendo o arquivo principal, o Makefile e as TAD's implementadas (pares `.h` `.c`) para solucionar o problema. Serão avaliados não só o resultado final do sistema mas também a identificação e documentação interna.

### 2 Descrição do Problema:

O “Júpiter Not Web - JNW” é um sistema offline de cadastro e matrícula de alunos em disciplinas da Universidade de São Paulo. Como um projeto ainda conceito, seu objetivo é proporcionar aos alunos da graduação uma matrícula rápida e segura além de desafogar o tráfego nos servidores da universidade comuns em épocas de matrícula.

Como todo projeto conceito, é necessário desenvolver um primeiro protótipo e apresentá-lo aos responsáveis para que os mesmos possam optar pela implementação ou não do sistema. Você é o programador responsável pela implementação do núcleo deste sistema, o futuro do “Júpiter Not Web” está em suas mãos.

Assim, elabore um sistema que possibilite o cadastro e remoção de alunos e disciplinas bem como a matrícula dos alunos. O sistema deverá trabalhar com 2 arquivos binários distintos, um deles para armazenar as informações dos alunos e outro para armazenar informações das disciplinas. O motivo para que estes arquivos sejam binários é manter a segurança da informação, pois estes não são legíveis em editores de texto básico.

Cada aluno é identificado unicamente pelo seu número USP e tem como informações importantes ao sistema o seu nome e ano de ingresso. Estas informações devem ser armazenadas em uma estrutura (ou `struct`) que será salva no arquivo binário ordenado pelo número USP. Similarmente, cada disciplina é identificada pelo seu código e tem como informações adicionais o nome da disciplina e o nome do professor responsável. Essas informações também devem ser armazenadas em uma estrutura (ou `struct`) e salvas no arquivo ordenadas pelo código.

A principal funcionalidade do sistema é efetuar a matrícula dos alunos, para isso trabalharemos com um arquivo de texto simples que deverá conter o número USP e o código da disciplina, fazendo assim uma associação de qual aluno está matriculado em qual disciplina.

### 3 Comandos do Sistema:

Como estamos desenvolvendo apenas um protótipo como prova de conceito, o sistema deverá funcionar como um terminal onde se aceita comandos de modo texto e o sistema reagirá de acordo com a especificação de cada comando:

- **sair** - Este comando deverá sair do sistema liberando toda memória possivelmente alocada e fechado corretamente os arquivos gerados;
- **cadastrar aluno** - Este comando deverá cadastrar as informações deste novo aluno, no arquivo binário, ordenado pelo número USP. As informações necessárias para este cadastro são primeiro nome (máximo 30 caracteres), número USP (7 caracteres) e ano de ingresso sendo essas informações fornecidas ao sistema necessariamente nesta ordem;
- **cadastrar disciplina** - Este comando deverá cadastrar as informações da disciplina ordenadamente no arquivo de disciplinas. As informações necessárias para este cadastro são nome da disciplina (máximo 30 caracteres), código da disciplina (7 caracteres), professor responsável (máximo 30 caracteres) que serão fornecidas ao sistema necessariamente nesta ordem;
- **matricular** - Este comando deverá salvar em um arquivo texto o número USP seguido do código da disciplina que o aluno está matriculado. O arquivo deverá ser salvo ordenadamente pelo número USP e caso o mesmo aluno esteja matriculado em mais de uma disciplina, deve estar ordenado entre eles pelo código da disciplina. Para fazer a matrícula, é necessário que exista um aluno e uma disciplina com aquele número USP e código respectivamente, caso um dos dois não exista, nada deve ser feito. Outro detalhe importante está no tamanho dos códigos e do número USP, por definição eles devem ter 7 caracteres, assim, caso tenham um tamanho menor que 7, deve ser preenchido o resto dos caracteres com espaço em branco, por exemplo, a matrícula do número USP 4 na disciplina Sc23 deve ser escrita no arquivo da seguinte maneira (ignorando-se as aspas):

```
'4      - Sc23  '
```

Aconselha-se salvar no arquivo adotando o seguinte padrão.

```
("%s - %s\n",numeroUSP,codigoDisc)
```

- **dump aluno** - Este comando deverá imprimir a lista ordenada de alunos de baixo para cima, ou seja, deve-se posicionar o ponteiro do arquivo no fim e ler os registros do ultimo para o primeiro. Para o correto funcionamento do sistema de submissão utilize da seguinte função para impressão dos registros de aluno:

```
printf("%d - %s - %d\n",reg.nusp, reg.nome, reg.ano);
```

- **dump disciplina** - Este comando deverá imprimir a lista ordenada de disciplinas de baixo para cima, ou seja, deve-se posicionar o ponteiro do arquivo no fim e ler os registros do ultimo para o primeiro. Para o correto funcionamento do sistema de submissão utilize da seguinte função para impressão dos registros de disciplinas:

```
printf("%s - %s - %s\n",reg.codigo, reg.nomeDisciplina, reg.professor);
```

- `dump matricula` - Este comando deverá imprimir a lista ordenada de matrículas de cima para baixo. Para o correto funcionamento do sistema de submissão utilize da seguinte função para impressão das matrículas:

```
printf("\%s - \%s\n", numeroUSP, codigoDisc);
```

## 4 Observações importantes:

- Os arquivos de alunos, disciplinas e matriculas devem ser aberto no inicio do programa e somente fechado no final da execução do programa, ou seja, não é permitido abrir e fechar os arquivos no meio da execução.
- Não é necessário garantir a coerência dos dados, ou seja, não serão adicionados dois alunos com o mesmo número USP e nem serão adicionados duas disciplinas com o mesmo código;
- Não poderá haver uma estrutura em memória interna, um vetor ou qualquer tipo de abstração que armazene na memória do computador todo o conteúdo do arquivo. Assim, para executar as funções de dump, busca e ordenação, deve-se explorar os dados diretamente no arquivo utilizando apenas de pequenos buffers com espaço para no máximo 2 registros de aluno, 2 registros de disciplina e/ou 2 linhas de matrícula.
- O nome dos alunos, nome da disciplina e professor responsável não terão espaço em branco no meio deles e nem utilizaram acentos gráficos. Por exemplo, a disciplina Computação Gráfica será inserida como “Computacao.Grafica”.
- Crie as TAD’s disciplina, aluno e matricula em que respectivamente deverão conter as funções para manipular o arquivo de disciplina, aluno e matricula.

## 5 Exemplo:

### Entrada:

```
cadastrar aluno
Felipe
6426830
2008
cadastrar aluno
Karina
6784034
2009
cadastrar disciplina
Comp.Grafica
SCC2320
Fernando
cadastrar disciplina
Org.Arquivos
SCC3203
Moacir
matricular
6426830
SCC2320
matricular
6784034
SCC3203
dump aluno
dump disciplina
dump matricula
sair
```

### Saída:

```
6426830 - Felipe - 2008
6784034 - Karina - 2009
SCC2320 - Comp.Grafica - Fernando
SCC3203 - Org.Arquivos - Moacir
6426830 - SCC2320
6784034 - SCC3203
```