

## Regulamento do trabalho final (leia com atenção)

### Do objetivo

O objetivo deste projeto é fomentar o trabalho em equipe e explorar temas avançados de programação *Assembly* e/ou paralela nas arquiteturas x86-64 e MIC.

### Dos grupos

O trabalho poderá ser realizado em grupo. Os grupos deverão possuir no máximo 2 integrantes e estes deverão ser preferencialmente da mesma turma. Os grupos devem ser postados no fórum do trabalho final até o dia **17/06/2015** juntamente com o tema escolhido e o repositório a ser usado.

### Dos temas

Os temas deverão ser escolhidos até o dia **17/06/2015** pelos grupos e postados no fórum do trabalho final. Cada grupo deverá implementar **no mínimo uma versão paralela** do algoritmo de Mandelbrot e comparar com a versão de referência fornecida pelo professor (<https://github.com/menotti/arg2asm/blob/master/pbm/mandelbrot.c>). Sugestões de temas (cada grupo pode escolher um ou mais):

- Acelerar o algoritmo por meio de instruções SIMD usando *inline assembly*;
- Acelerar o algoritmo usando autovetorização;
- Comparar códigos gerados por diferentes compiladores e parâmetros de compilação;
- Comparar desempenho em diferentes processadores/arquiteturas;
- Implementar o algoritmo na arquitetura MIC (não é necessário usar assembly);
- Implementar uma versão gráfica interativa do algoritmo (dando zoom a partir do clique);
- Modificar e acelerar o algoritmo em ponto fixo (converter para inteiros).

Para todos os temas é necessário explicar a razão do ganho de desempenho, estudando e demonstrando as diferenças.

### Da entrega

Todos os trabalhos deverão ser postados até o dia **24/06/2015** em um repositório de software usando um sistema de controle de versões (Bitbucket, GitHub, GoogleCode, SourceForge, etc). Aconselha-se que o grupo faça atualizações periódicas no repositório, evitando deixar tudo para última hora, bem como atualizações com pouco conteúdo.

As entregas devem ser feitas simultaneamente em duas formas:

1. Códigos-fonte C/C++ com *inline assembly* usados na implementação (não enviar arquivos gerados pelas ferramentas, apenas os fontes). O código deve vir acompanhado de um Makefile ou arquivo make.bat que:
  - i. Compile o código-fonte, gerando os executáveis (original e modificados);
  - ii. Execute os programas, gerando arquivos de saída, cujo sufixo do nome indique a técnica de paralelização usada (ex. “\_simd\_x.pnm”, onde x é conjunto de instruções usado (mmx, sse, avx, etc)). No caso de versões interativas não é necessário gerar arquivos de saída, apenas incluir um indicativo de desempenho (por exemplo: fps).

2. Wiki no próprio repositório documentando e apresentando o trabalho realizado (não é preciso preparar diapositivos para a apresentação). A página deve conter exemplos de imagens geradas, vídeos e gráficos de comparação de desempenho entre as versões.

## **Da apresentação**

Não haverá apresentação dos trabalhos em aula. Os alunos deverão apenas postar, no fórum do trabalho final, o link do repositório para avaliação. O professor não irá avaliar entregas que não geram os resultados a partir do *make*.

## **Dos critérios de avaliação**

Os grupos serão avaliados seguindo os seguintes critérios:

- Clareza e qualidade do código desenvolvido (25%): incluir comentários que ajudem a compreender a solução, bem como usar técnicas consagradas de desenvolvimento de software;
- Qualidade da documentação e da apresentação (wiki) (25%): gerar uma documentação que demonstre claramente o que foi feito e qual foram os resultados obtidos. É fundamental explicar os resultados, dizer que x foi melhor que y não ajuda muito se não for apresentada uma justificativa.
- Balanceamento do trabalho entre os membros do grupo (25%): o trabalho é em grupo, não deve ser feito por um único membro. A avaliação será individual, a partir das contribuições no repositório e no fórum.
- Comunicação/contribuições com/para outros grupos no fórum (25%): todas as mensagens trocadas no fórum poderão contribuir para a nota, desde comandos do controle de versão até dúvidas dos compiladores e instruções.

## **Do plágio**

Considera-se plágio o uso de qualquer fragmento de código que não tenha sido desenvolvido na disciplina. Não se considera plágio exemplos de código encontrados na documentação oficial da Intel ou da Microsoft, desde que devidamente referenciados. Quaisquer projetos que sejam considerados plagiados serão anulados. A troca de experiências e informações entre os grupos deve ser feita no fórum do trabalho e não será considerada plágio.