

Automatização do processo de obtenção e transmissão de dados

- **Objetivo:** O aluno deverá se familiarizar com alguns comandos em linguagem bash e protocolos de comunicação para viabilizar a obtenção de dados, processamento, geração e divulgação dos produtos de forma automatizada.
- **Missão:** Instalamos um transmissor em uma baleia que emite sinais com as seguintes informações: posição, tempo da primeira medição, profundidade e temperatura da água. **A sua missão é manter um arquivo com todas as informações transmitidas, fazer um controle de qualidade para remover dados espúrios ou repetidos e gerar um gráfico com a posição da baleia, juntamente com os registros de temperatura e profundidade.** Os gráficos deverão conter dados da série completa da trajetória da baleia, não somente do último arquivo. A figura deverá mostrar o tempo da primeira e última coleta de dados, bem como a data que o gráfico foi gerado.

Primeiramente, algumas explicações sobre os dados:

- Os arquivos com os dados transmitidos pelo sensor estão disponíveis no site:
<http://www3.io.usp.br:32080/los/IOF0265/baleia/>
- A cada 5 minutos, um novo arquivo é transmitido e disponibilizado no site. Somente os 10 arquivos mais recentes estão disponíveis. Ou seja, a cada 5 min. o arquivo mais antigo é removido permanentemente do site.
- O nome do arquivo é composto pela data do primeiro registro e escrito no seguinte formato: data_file_YYYYMMDD_HHMMSS.txt, onde YYYY é o ano, MM o mês, e assim por diante.
- A primeira linha do arquivo mostra um cabeçalho. O arquivo contém a longitude, latitude, profundidade (m) e temperatura (°C).

Guia do que precisa ser feito:

1. Traga os arquivos via browser ou wget para analisar o conteúdo dos mesmos. Na primeira vez o download pode ser feito manualmente, mas se preferir já pode começar a implementar um script em bash (`baleia_script.sh`) para trazer todos os arquivos do site usando wget, por exemplo. Lembre-se que a primeira linha de um bash script começa com `#!/bin/bash`;
2. Para cada um dos arquivos, remover a linha do cabeçalho. Para fazer isso, fornecemos as seguintes linhas de bash script:

```
for file in $(ls -1 data_file*);do
name=$(echo $file|sed s/data_file/track/g)
tail -n +2 $file > $name
done
```
3. Mantenha um diretório com todos os dados originais e um outro com os arquivos modificados. Estes últimos serão os que vão ser processados. Lembre que a cada 5 minutos um novo arquivo é adicionado ao site. Você deverá manter um banco de dados com todos os arquivos transmitidos. Modifique o script acima e mova os arquivos para os seus respectivos diretórios.
4. Escreva um programa para ler, processar e plotar os gráficos. A figura abaixo é um modelo de como gostaríamos que os gráficos ficassem porém estamos revelando somente o da posição.
5. Aqui está uma mãozinha para vocês reconstruírem o vetor tempo:

```
onesec=datenum(0,0,0,0,0,1)-datenum(0,0,0,0,0,0);
t0=datenum(yy0,mm0,dd0,ho0,mi0,se0);
tvec=[0:length(lo)-1]*onesec;
t=t0+tvec;
```

Lembre-se que o tempo inicial está escrito no nome do arquivo. As variáveis que terminam com zero, darão o tempo inicial. Como cada linha do arquivo é gerado em 1 segundo, esses comandos ajudarão a reconstruir o vetor tempo corretamente.

6. Os dados precisam passar por um controle de qualidade. Observe os “flags” de dado ruim na profundidade e na temperatura. Adicionalmente, notem que há uma repetição de dados no tempo: o último dado de um arquivo é repetido como sendo o primeiro no próximo arquivo. Inclua na rotina um condição para remover o dado repetido.

7. Após terminar a rotina de processamento com a geração do gráfico, faça as rotinas para executar os programas automaticamente. Siga o seguinte roteiro:

(a) Crie um programa em bash script (e.g. `bash_mat.sh`) com o seguinte conteúdo:

```
#!/bin/bash
nohup /caminhodomatlab/matlab -nosplash -display null < $1
```

(b) Não se esqueça de tornar esse script executável: `chmod +x bash_mat.sh`.

(c) Agora pegue aquele bash script que você criou no item 1. Inclua os comandos para executar o seu programa através de um bash script:

```
bash_mat.sh plot_programa.m
```

(d) Execute o seu bash script na linha de comando para verificar se funciona.

(e) Finalmente, coloque o programa para ser executado no seu computador usando o comando `crontab -e` (-e é opção para editar e provavelmente o editor conhecido como vi, o melhor editor que existe, será iniciado). Inclua as seguintes linhas no crontab:

```
# m h dom mon dow    command
*/5 * * * * baleia_script.sh
Digite man crontab para saber mais a respeito.
```

(f) Salve e saia do crontab e voila! Se tudo der certo, você terá uma figura atualizada a cada 5 minutos.

8. Se não der certo na primeira tentativa, não se desespere. Fazer um bash script requer disciplina. Todos os programas e rotinas que são chamados devem estar declarados com o caminho completo, senão não funciona.

