

Laboratório de Sistemas Operacionais Embarcados (ECOS13)

Prof Otávio Gomes (otavio.gomes@unifei.edu.br)

Atividade de verificação

Device driver controller

1) [35%] A partir da análise dos códigos-fonte disponíveis nos arquivos **ECOS13_Lab03_main.c** e **ECOS13_Lab03_driver.h** faça:

- a) Implemente a verificação da variável `func_id` dentro de um intervalo válido para o array de funções do driver. Isso aumentará a segurança do código, impedindo acessos fora dos limites do array, que podem causar um comportamento indefinido.
- b) Implemente a possibilidade de alocação de outras funções em cada um dos grupos de dispositivos/ações do driver. Neste cenário é necessário realizar o gerenciamento de arrays de ponteiros para funções de tamanho variável. Para isso, crie três funções adicionais para cada uma das instâncias apresentadas no código:

```
ptrFunc genFuncs[] = {initGenericoDriver};  
ptrFunc intFuncs[] = {initInterruptDriver};  
ptrFunc timFuncs[] = {initTimerDriver};
```

Não se esqueça da utilização de `enum` e da chamada das funções que deverá estar presente na `main`: `callDriver(X, Y, NULL)`;

Laboratório de Sistemas Operacionais Embarcados (ECOS13)

Prof Otávio Gomes (otavio.gomes@unifei.edu.br)

- 2) Controladora de drivers - A partir das modificações e melhorias anteriores, proponha uma modificação no código que deverá:
- a) [25%] Executar todas as funções criadas pelo menos uma vez. Faça uma generalização da chamada `callDriver`, utilizando um laço para varredura de um buffer circular, por exemplo.
 - b) [25%] Crie um retorno de funções que permita à função ser executada novamente, caso solicite, conforme um número de vezes pré-definido. Realize as modificações necessárias no código para que este retorno seja aplicável. Por exemplo: `#define RETORNA 2`
 - c) [25%] Modifique as funções **`callDriver`** e **`main`** de modo que as funções que solicitarem novo acesso e execução sejam inseridas novamente na fila de processos.