

# RELATÓRIO

## P1: TROCAS DE CONTEXTO

Aluno: Gabriel Claudino de Souza - GRR20215730

### 1. Explique o objetivo e os parâmetros de cada uma das quatro funções acima.

R:

**getcontext(&a):** Objetivo: salvar o contexto atual na variável 'a'; Parâmetro: ponteiro para uma variável do tipo ucontext\_t onde o contexto atual será salvo.

**setcontext(&a):** Objetivo: restaurar um contexto salvo anteriormente na variável 'a'; Parâmetro: ponteiro para uma variável do tipo ucontext\_t que foi previamente salva.

**swapcontext(&a, &b):** Objetivo: salva o contexto atual em 'a' e muda para o contexto 'b'; Parâmetros: '&a': ponteiro para onde o contexto atual será salvo; '&b': ponteiro para o contexto que será desviado a execução.

**makecontext(&a, ...):** Objetivo: preparar um novo fluxo de execução previamente criado e executar uma função quando for chamado(setcontext() ou swapcontext()); Parâmetros: '&a': ponteiro para contexto já inicializado; outros parâmetros: função a ser executada; número de argumentos da função; argumentos da função.

### 2. Explique o significado dos campos da estrutura ucontext\_t que foram utilizados no código.

R:

**uc\_stack.ss\_sp = stack:** ponteiro para o início da pilha que o contexto utilizará.

**uc\_stack.ss\_size = STACKSIZE:** tamanho da pilha do contexto.

**uc\_stack.ss\_flags = 0:** flags para controle da pilha, no caso 0 será o padrão.

**uc\_link = 0**: próximo contexto que será executado quando o atual terminar, no caso 0, o contexto termina.

**3. Explique cada linha do código de contexts.c que chame uma dessas funções ou que manipule estruturas do tipo ucontext\_t.**

R:

Variáveis Globais(Linha 20):

**ContextPing**: contexto da função BodyPing();

**ContextPong**: contexto da função BodyPong();

**ContextMain**: contexto da main.

Função BodyPing():

**swapcontext(&ContextPing, &ContextPong)**(Linha 33): Salva o contexto atual em 'ContextPing' e muda para o contexto 'ContextPong'.

**swapcontext(&ContextPing, &ContextMain)**(Linha 37): Ao terminar BodyPing(), troca para o contexto da main.

Função BodyPong():

**swapcontext(&ContextPong, &ContextPing)**(Linha 51): Salva o contexto atual em 'ContextPong' e muda para o contexto 'ContextPing'.

**swapcontext(&ContextPing, &ContextMain)**(Linha 55): Ao terminar BodyPong(), troca para o contexto da main.

Na main():

**getcontext(&ContextPing)**(Linha 66): Salva o contexto da main em 'ContextPing' para configurar em **makecontext(&ContextPing, (void\*)(\*BodyPing), 1, " Ping")**(Linha 82), onde prepara o 'ContextPing' para executar a função **BodyPing()** quando 'ContextPing' for ativado.

**getcontext(&ContextPong)**(Linha 84): Salva o contexto da main em 'ContextPong' para configurar em **makecontext(&ContextPong, (void\*)(\*BodyPong), 1, " Ping")**(Linha 100), onde prepara o

'ContextPong' para executar a função **BodyPong()** quando 'ContextPong' for ativado.

**swapcontext(&ContextMain, &ContextPing)**(Linha 102): Salva o estado da main em 'ContextMain' e começa a executar 'ContextPing'.

**swapcontext(&ContextMain, &ContextPong)**(Linha 103): Salva o estado da main em 'ContextMain' e começa a executar 'ContextPong'.

**4. Para visualizar melhor as trocas de contexto, desenhe o diagrama de tempo dessa execução.**

