

# Relatório sobre *contexts.c*

Waine Junior

Giovanni Forastieri

Agosto de 2019

## 1 Introdução

Este relatório tem como objetivo explicar a execução do código presente no arquivo *contexts.c*. Primeiro descrevendo os objetivos e parâmetros das funções referentes a contextos presentes nele. Depois a estrutura de dado utilizada. Por fim o que cada linha de código que manipulam contextos fazem. Além disso, é apresentado o diagrama de tempo da execução do código.

## 2 Funções

O código utiliza funções padrão POSIX para manipulação de contexto. São elas:

- `getcontext (&a)`: tem como objetivo obter o atual contexto do programa e gravá-lo na variável `a`, a qual é do tipo `ucontext_t`, estrutura explicada na seção 3.
- `setcontext (&a)`: tem como objetivo alterar o contexto atual, restaurando aquele apontado pela variável `a`.
- `swapcontext (&a, &b)`: tem como objetivo trocar o contexto, salvando o atual em `a` e restaurando aquele em `b`. É equivalente à chamada de `getcontext (&a)` ; seguida de `setcontext (&b)` ;.
- `makecontext (&a, ...)`: tem como objetivo alterar o contexto salvo em `a`, mais especificamente a "função chamada" por esse e os argumentos passados para essa.

## 3 Estruturas de dados

Para o armazenamento das propriedades do contexto, é utilizada uma estrutura de dado presente no arquivo *ucontext.h*, denominada `ucontext_t`. O significado dos campos da estrutura utilizados no código é:

- `void* uc_stack.ss_sp`: *stack pointer* da pilha de sinal (*signal stack*).
- `size_t uc_stack.ss_size`: tamanho, em bytes, da pilha de sinal. Deve ser definido como o tamanho alocado para a pilha. O arquivo *signal.h* define um valor padrão para ser utilizados: `SIGSTKSZ`, tamanho canônico.

- `int uc_stack.ss_flags`: operador ou lógico entre as *flags* `SS_DISABLE` e `SS_ONSTACK`. A primeira diz para o sistema se a pilha de sinal não deve ser utilizada. A segunda é uma variável setada pelo sistema que diz se a pilha está em uso atualmente. Caso não esteja em uso, os sinais devem ser entregues à pilha do usuário normal
- `ucontext_t.uc_link`: aponta para o contexto que será resumido após o fim da execução do contexto da estrutura.

A pilha de sinal (*signal stack*) é utilizada para definir o tratamento dos sinais gerados durante a execução de um programa, por meio da função `signal()` ou `sigaction()`. O levantamento de um sinal pode ser feita por meio das chamadas `kill()` e `raise()`. Os tipos padrão de sinais que podem ser gerados são definidos no arquivo *signal.h* [1].

## 4 Código

## 5 Diagrama de tempo da execução

## Referências

- [1] GNU. *GNU C Library*, August 2019. Disponível em: [https://www.gnu.org/software/libc/manual/html\\_node/](https://www.gnu.org/software/libc/manual/html_node/).