

NOME: Victor Dallagnol Bento - 201520835
DISCIPLINA: Sistemas Operacionais em Tempo Real
PROFESSOR: Osmar Marchi dos Santos
DATA: 02 /07/2018



TRABALHO FINAL

INTRODUÇÃO

O sistema tem como ideia principal simular um hardware que faz o cálculo de *IMC* - *Índice de Massa Corpórea* em adultos. No monitor, os dados são coletados através de sensores que captam a altura e o peso (via terminal), eles são colocados em um pacote (buffer) que é enviado para o simulador. No simulador, além do cálculo do IMC são calculados os valores médios das alturas e dos pesos dos indivíduos, assim como a tendência do indivíduo (uma estimativa). Por meio do monitor (através de comandos) será possível escolher entre as 3 opções de funções, todos os dados serão calculados e estimados por funções periódicas, e posteriormente estes dados serão salvos em um *buffer*, que será enviado ao *monitor*.

DESCRIÇÃO E IMPLEMENTAÇÃO

o Tarefas Periódicas:

– calcula_IMC:

Função que efetua o cálculo do IMC. Se o peso e a altura não forem nulos efetua o cálculo do *imc*, e converte o mesmo para *char*. Ocorre a atualização do numero de alturas e pesos lidos, assim como a soma para as variáveis que farão as médias.

Por fim, o valor já convertido do *imc* é salvo no buffer de informações *save*. Após isso a linha do buffer *save* (variável global) é incrementada e a variável que indica se o buffer está vazio (*vazio*) é atualizada.

A periodicidade da tarefa é controlada pela função:

```
usleep(rand()%TIME_IMC);
```

Que é controlada pela definição *TIME_IMC* definida no inicio do código através de um *#define*.

– medidas_medias:

Função que efetua o calculo das médias dos pesos e alturas coletados até o momento. Se o comando indicado for 1 e o peso e a altura não forem nulos ocorre a atualização do numero de alturas e pesos lidos, assim como a soma para as variáveis que farão as médias. Em seguida, ocorre a o cálculo específico das médias e sua conversão para o tipo *char*, salvando assim, o resultado final no buffer *save*.

A periodicidade da tarefa é controlada pela função:

```
usleep(rand()%TIME_MED);
```

Que é controlada pela definição *TIME_MED* definida no inicio do código através de um *#define*.

– tendencia:

Efetua o cálculo das tendências considerando 10% do nível do IMC (magreza, normalidade, sobrepeso e obesidade). Esta função necessita do valor do IMC para efetuar sua estimativa, por este fato, para que a função execute é necessário que o comando seja 2 e que o valor do *imc* não seja nulo. A função também atualiza o numero de alturas e pesos lidos, assim como a soma para as variáveis que farão as

médias. Feito isso, considerou-se 10% do valor do IMC para que assim fosse possível comparar por meio de *if* e *else* o valor da estimativa do indivíduo. Calculada a estimativa, o resultado é salvo no buffer *save*.

A periodicidade da tarefa é controlada pela função:

```
usleep(rand()%TIME_TEND);
```

Que é controlada pela definição *TIME_TEND* definida no início do código através de um *#define*.

o Comandos:

- 0, altura, peso - Informa o Índice de Massa Corpórea do indivíduo;
- 1, altura, peso - Informa valores médios de peso e altura do grupo de indivíduos;
- 2, altura, peso - Informa tendência do indivíduo;

o Mutex:

No código foram utilizado dois *mutex*: *protect* e *mutex*. O primeiro é utilizado para modificações das variáveis globais assim como a para escrita e leitura do *buffer* de informações, utilizado nas funções *calcula_IMC*, *medidas_medias*, *tendencia*, e *send_results*. O segundo é utilizado para criação das *threads* das funções periódicas e para a escrita no *socket* (comunicação), utilizado na função *send_results*.

o Variável de Condição:

É utilizada uma flag de controle (*vazio*) indicando se o *buffer* contém ou não informações. A variável de condição *cond* é utilizada dentro de um laço *while*, na função *send_results*, esperando até que o *buffer* contenha alguma informação.

o Comunicação:

A comunicação entre monitor e simulador ocorre por meio das funções:

<i>send(sockfd,buffer,50,0);</i>	monitor → simulador
<i>recv(sockfd,buffer,50,0);</i>	simulador → monitor
<i>write(newsockfd,buffer,50);</i>	simulador → monitor
<i>read(newsockfd,buffer,50);</i>	monitor → simulador

o Esquemático:

