Support Library – Version 2017/1

1 Descrição Geral

A biblioteca *support* é composta por várias funções que auxiliam (dão suporte) na implementação dos trabalhos de Sistemas Operacionais IN dos cursos de Ciência da Computação e Engenharia de Computação do Instituto de Informática da UFRGS.

Atualmente estão disponíveis três grupos de função:

- Uma função para obter a versão atual da biblioteca;
- Funções para tratamento de filas;
- Uma função que fornece um número pseudo-aleatório de 8 bits, sem sinal.

As filas são estruturas de dados importantes em um sistema operacional e servem, entre outros usos, para organizar as estruturas de dados que representam os processos, existentes em um escalonador de curto prazo. Assim, por exemplo, sempre que um processo realiza uma transição de estados, o escalonador passa as estruturas de dados correspondentes ao processo de uma fila para outra.

2 Interface de Programação da biblioteca

A biblioteca utiliza duas estruturas de dados: "*sFilaNode2*" e "*sFila2*", que estão definidas no arquivo *support.h*, junto com a declaração dos tipos "NODE2" e "PNODE2", para a primeira estrutura, e "FILA2" e "PFILA2", para a segunda estrutura, respectivamente.

A estrutura "*sFila2*" possui os elementos necessários para representar uma fila. Os elementos "*first*" e "*last*" apontam para uma estrutura do tipo "*sFilaNode2*", que representam o primeiro e o último nodo da fila, respectivamente. O elemento "*it*" é o "iterador" da fila, usado para "navegar" pelos elementos da mesma (ver funções *FirstFila2*, *LastFila2* e *NextFila2*).

A estrutura "*sFilaNode2*" possui os elementos necessários para representar um **item** (nodo) da fila. Nessa estrutura estão os ponteiros "*next*" e "*ant*", usados para ligar um nodo aos seus vizinhos "seguinte" e "anterior", respectivamente. Também está presente um ponteiro "*node*", usado para salvar o ponteiro dos dados armazenados nesse item da fila.

As estruturas de dados, conforme aparecem no arquivo "*support.h*", são apresentadas na figura 1.

O detalhe das funções da biblioteca podem ser encontrados no arquivo "*support.h*", inclusive o significado dos códigos de retorno das funções.

```
struct  sFilaNode2 {
        void    *node;                  // Ponteiro para a estrutura de dados do NODO
        struct  sFilaNode2 *ant;        // Ponteiro para o nodo anterior
        struct  sFilaNode2 *next;       // Ponteiro para o nodo posterior
};
struct sFila2 {
        struct  sFilaNode2 *it;         // Iterador para varrer a lista
        struct  sFilaNode2 *first;      // Primeiro elemento da lista
        struct  sFilaNode2 *last;       // Último elemento da lista
};

typedef struct sFilaNode2       NODE2;
typedef struct sFila2           FILA2;
typedef struct sFilaNode2 *     PNODE2;
typedef struct sFila2 *         PFILA2;
```

Figura 1 – Estruturas de dados que devem ser usadas na implementação

2.1 Descrição da Interface

int **Version** (void)

Retorna número que identifica a versão da biblioteca. A versão é calculada da seguinte forma: (2 * ANO) + (SEMESTRE – 1)

unsigned char **Random2**(void)

Gera um número pseudo-aleatório entre 0 e 255

int **CreateFila2** (PFILA2 *pFila*)

Inicializa uma estrutura de dados do tipo FILA2.

int **FirstFila2** (PFILA2 *pFila*)

Posiciona o iterador da fila no primeiro item da mesma.

int **LastFila2** (PFILA2 *pFila*)

Posiciona o iterador da fila para o último item da mesma.

int **NextFila2** (PFILA2 *pFila*)

Posiciona o iterador da fila para o item seguinte àquele apontado pelo iterador antes da chamada da função.

void ***GetAtIteratorFila2** (FILA2 *pFila*)

Retorna o ponteiro armazenado no conteúdo do item endereçado pelo iterador da fila. É o elemento "nodo" da estrutura "sFilaNodo2".

int **AppendFila2** (PFILA2 *pFila*, void **content*)

Salva o ponteiro "*content*" em um novo item e coloca-o no final da fila "*pFila*". Requer alocação dinâmica da estrutura "sFilaNodo2".

int **InsertAfterIteratorFila2** (PFILA2 *pFila*, void **content*)

Coloca o ponteiro "*content*" em um novo item e coloca-o logo após o item apontado pelo iterador da fila "*pFila*". Requer alocação dinâmica dessa estrutura "sFilaNodo2".

int **DeleteAtIteratorFila2** (PFILA2 *pFila*)

Remove da fila "*pFila*" o item indicado pelo iterador e libera a memória correspondente à estrutura "sFilaNodo2". Ao final da função, o iterador deverá estar apontando para o elemento seguinte àquele que foi apagado. Vai receber NULL se a fila ficar vazia.

2.2 Dicas de Utilização

Algumas dicas e detalhes de utilização da biblioteca:

1. Para criar uma fila é necessário declarar uma variável do tipo FILA2 e inicializar seu conteúdo, usando a função "*CreateFila2*".
2. Essa biblioteca é útil para armazenar filas de quaisquer tipos de dados. Isso é possível porque os dados a serem armazenados devem ser passados na forma de ponteiros do tipo (*void **). Ao inserir um item na fila deve-se realizar o *type-casting* para (*void **) e, ao ler um item da fila deve-se realizar o *type-casting* para a estrutura correta do item obtido.
3. A implementação da fila armazena apenas os ponteiros fornecidos. Ela não armazena os dados apontados. A área com esses dados deve ser mantida alocada pela aplicação que utiliza a biblioteca.

Apesar deste documento empregar o nome "*support library*" e mencionar o termo *biblioteca*, as funções são oferecidas na forma de um arquivo binário. Formalmente, nos ambientes Unix e C, uma biblioteca é um tipo especial de arquivo com uma extensão *.a* (bibliotecas estáticas) e *.so* (bibliotecas dinâmicas). Os sistemas Windows também possuem bibliotecas estáticas e dinâmicas, essas últimas são mais conhecidas pelo acrônimo DLL (*Dynamic Link Library*). Neste documento, o termo biblioteca foi livremente empregado para fazer alusão a "um conjunto de funções".

Para utilizar as funções de *support.o* em seus programas é necessário ligar o arquivo *support.o* com os demais binários e bibliotecas que comporão o arquivo executável final. Essa ligação pode ser feita, por exemplo, com o seguinte comando:

gcc -o myprog.c support.o

Esse exemplo realize a compilação do programa myprog.c considerando que ele utilize uma ou mais funções providas pelo binário *support.o*. A execução do *gcc* com a opção *–o* gera o arquivo executável *myprog* e realiza a ligação com *support.o*, adicionando a esse executável o corpo das funções de *support.o*.