

## Ficheiro Asm\_utils.S

Este ficheiro contém duas funções, `copyBuffer` e `cleanBuffer`:

- **copyBuffer**: em C, teria uma declaração semelhante a

```
void copyBuffer(char* bf1, char* bf2, int size);
```

Esta função serve como uma alternativa a `memcpy()` e copia os `size` primeiros bytes de `bf2` para `bf1`. Faz uso das instruções de tratamento de sequência `rep movsb` e é utilizada na implementação do double buffering.

- **cleanBuffer**: em C, teria uma declaração semelhante a

```
void cleanBuffer(char bf1, int size, short color);
```

Esta função serve como uma alternativa a `memset()` e preenche os `size` primeiros bytes de `bf1` com a cor `color`. Faz uso das instruções de tratamento de sequência `rep stosw` e é também utilizada na implementação do double buffering.

## Ficheiro kbd\_read\_asm.S

Este ficheiro contém uma função, `kbd_read_asm`:

- **kbd\_read\_asm**: em C, teria uma declaração semelhante a

```
char kbd_read_asm();
```

Esta função lê um byte do output buffer do teclado, verificando a existência de error, e retorna-o. Se houver um erro, retorna 0.

## Ficheiro rtc\_read\_asm.S

Este ficheiro contém uma função, `rtc_read_asm`:

- **rtc\_read\_asm**: em C, teria uma declaração semelhante a:

```
char rtc_read_asm(char addr);
```

Esta função escreve o endereço que lhe é passado como argumento para o `RTC_ADDR_REG`, e de seguida lê o conteúdo do registo `RTC_DATA_REG`, retornando o valor dessa leitura.