

ORIENTAÇÕES PARA TESTE DO SEU KERNEL

Para usar a APP desenvolvida pelo professor

- 1) Realize a montagem do seu kernel
- 2) Carregue no simulador o arquivo “app_prof_sem_kernel.mem”
- 3) Realize a carga parcial de seu kernel, usando os seguintes endereços: 32768 65535 32768

Para usar uma APP diferente (por exemplo, as APPs individuais fornecidas), siga os procedimentos abaixo. Veja que a diferença em relação ao procedimento anterior é a montagem e a carga da nova APP.

- 1) Realize a montagem do seu kernel
- 2) **Realize a montagem da APP**
- 3) Carregue no simulador o arquivo “app_prof_sem_kernel.mem”
- 4) **Realize a carga parcial da aplicação, usando os seguintes endereços: 256 16383 256**
- 5) Realize a carga parcial de seu kernel, usando os seguintes endereços: 32768 65535 32768

DESCRIÇÃO DAS APLICAÇÕES ESPECÍFICAS DE TESTE

[1] putchar.ced

- Essa APP utiliza apenas a “putchar”.
- Ao rodar o programa, será escrito no visor a mensagem “INF 108”

[2] getchar.ced

- Essa APP utiliza apenas a “getchar”
- Ao rodar o programa, ele vai esperar que sejam digitadas duas teclas, e então vai encerrar.
- As teclas digitadas serão escritas nos endereços 1024 e 1025. Você deve verificar se os valores corretos estão escritos nesses endereços.

[3] putmsg.ced

- Essa APP utiliza apenas a “putmsg”
- Ao rodar o programa, será escrito no visor a mensagem “putmsg test!”

[4] setcursor.ced

- Essa APP utiliza a função “putchar”, para fins de teste da função “setcursor”
- A função “putchar” deve estar operando corretamente.
- Ao rodar o programa, será usada a função “setcursor” para posicionar o cursor e escrever os números em ordem numérica. Ao final, deve aparecer no visor a seguinte mensagem:
 - “6 4 2 5 3 1 0 7”

[5] timer.ced

- Essa APP utiliza a função “putchar”, para fins de teste das funções “start” e “ready”
- A função “putchar” deve estar operando corretamente.

- Ao rodar o programa, será apresentado no visor a seguinte mensagem “0123456789!”.
- Cada caractere será apresentado no visor com um tempo entre apresentações de 1 (um) segundo.

[6] kbhit.ced

- Essa APP utiliza as funções “getchar” e “putchar”, para fins de teste da função “kbhit”
- As funções “getchar” e “putchar” devem estar operando corretamente.
- Ao rodar o programa, deverá aparecer no visor o cursor, o que indica que o programa está aguardando alguma tecla. Essa espera está sendo realizada através do “kbhit”
- Ao ser digitada qualquer tecla, será apresentada no visor a letra “K”, seguida da tecla que foi digitada.

[7] bs.ced

- Essa APP testa a operação do “back space” (BS). Para isso, utiliza a função “putchar”
- Ao rodar o programa, é escrito no visor (usando a “putchar”) a seguinte sequência de caracteres: “123BSabc”, seguida de três caracteres BS (código H08), e três caracteres “-”.
- Ao final, deverá estar no visor, a partir da posição zero, apenas “123BS---”

[8] bs0.ced

- Essa APP testa a operação do “back space” (BS), quando chega na primeira posição à esquerda do visor. Para isso, utiliza a função “putchar”.
- Ao rodar o programa, é escrito no visor a sequência “123BSabc” (como no “bs.ced”), porém seguindo de 12 (doze) caracteres BS e, então, por três caracteres “---”.
- Ao final, deverá estar no visor, a partir da posição zero, os seguintes caracteres “---BSabc”

[9] putmsg_long.ced

- Essa APP testa a operação do “putmsg”, quando a mensagem é maior do que a capacidade do visor.
- Ao rodar o programa, é escrito no visor a seguinte mensagem: “putmsg long putmsg long ----->!!!!!!!!!!!!”. Essa mensagem ocupa 36 caracteres até o caractere “>”. Os caracteres “!” estão fora da área do visor.
- Ao final, deverá estar no visor, a partir da posição zero, apenas os seguintes caracteres: “putmsg long putmsg long ----->”.