**Lab 6 Stack Frame Practice**

Read the code in lab6.asm. Write the stack frame for strcpy:

|  |  |  |  |
| --- | --- | --- | --- |
| Address | content | details | notes |
| 0x21FF |  |  | In the main, the caller pushes the parameters onto the stack. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  | The return address (the address of the next command) is pushed onto the stack automatically when “call strcpy” is executed. |
|  |  |  |
|  |  |  |
|  |  |  | In the subroutine, the callee pushes those registers which will be used onto the stack to protect them from being changed. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 0x21F1 |  | **🡨 SP** |  |
|  | …….. |  |  |
|  |  |  |  |
|  |  |  |  |

Design the stack frame for strlength:

|  |  |  |  |
| --- | --- | --- | --- |
| Address | content | details | notes |
| 0x21FF |  |  | In the main, the caller pushes the parameters onto the stack. |
|  |  |  |
|  |  |  | The return address (the address of the next command) is pushed onto the stack automatically when “call strlength” is executed. |
|  |  |  |
|  |  |  |
|  |  |  | In the subroutine, the callee pushes those registers which will be used onto the stack to protect them from being changed. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 0x21F5 |  | **🡨 SP** |  |
|  | …….. |  |  |
|  |  |  |  |
|  |  |  |  |