1. instruction fetch (00000,FFC):
    - fault(00000,FFC, instruction fetch)
    - 00000[000].page = 00001
    - return_from_fault
    - instruction fetch (00000,FFC) (2nd attempt)
    - fault(00000,FFC, instruction fetch)
    - 00001[000].page = 00002
    - return_from_fault
    - instruction fetch (00000,FFC) (3rd attempt)
    - page allocate = 00002
2. attempt (PUSH #10):
    - SP = SP - 4 (08001,000 - 4 = 08000,FFC)
    - store(08000,FFC): Write the value 10 to the memory address (08000,FFC)
    - fault(08000,FFC, memory access)
    - 00000[020].page = 00003
    - return_from_fault
    - store(08000,FFC) (2nd attempt)
    - fault(08000,FFC, memory access)
    - 00003[000].page = 00004
    - return_from_fault
    - store(08000,FFC) (3rd attempt)
    - page allocate = 00004
    - success
3. instruction fetch (00001,000):
    - fault(00001,000, instruction fetch)
    - 00001[001] = 00005
    - return_from_fault
    - instruction fetch (00001,000) (2nd attempt)
    - page allocate = 00005
4. attempt (CALL 2,000):
    - SP = SP - 4 (08000,FFC - 4 = 08000,FF8)
    - store(08000,FF8): Write the return address (00001,004) to the memory address (08000,FF8)
    - page allocate = 00004
    - success
5. instruction fetch (00002,000):
    - fault(00002,000, instruction fetch)
    - 00001[002].page = 00006
    - return_from_fault
    - instruction fetch (00002,000) (2nd attempt)
    - page allocate = 00006
6. attempt (MOV EAX → *(10,000)):
    - load(00010,000): Read data from the memory address (00010,000)
    - fault(00010,000, memory access)
    - 00001[010].page = 00007
    - return_from_fault
    - load(00010,000) (2nd attempt)
    - page allocate = 00007

   - store(*(10,000)): Write the value at EAX to the memory address *(10,000)
   - success

7. instruction fetch (00002,004):
   - page allocate = 00006
8. attempt (HALT):
   - success