SAVING

```
; first row, first column  --- 1024
; second row, first column --- 1064
; third row, first column  --- 1104

; cl65 -o saving -t c64 -I ~/cc65/include -L ~/cc65/lib -C c64-asm.cfg-MINE saving.s
; sys49152

; using screen as memory locations

    ; ***** larry; data to be save; using PETSCII
    lda #12
    sta 1024
    lda #1
    sta 1025
    lda #18
    sta 1026
    lda #18
    sta 1027
    lda #25
    sta 1028

    ; ***** ABC; filename; using ASCII
    lda #65
    sta 1064
    lda #66
    sta 1065
    lda #67
    sta 1066

    ; ***** saving
    lda #$01        ; logical file number 1
    ldx #$09        ; device 9 (disk drive)
    ldy #$02        ; secondary address 2
    jsr $ffba       ; call SETLFS

    lda #3          ; length of filename
    ; 1064 = $0428
    ldx #$28        ; low byte of filename address
    ldy #$04        ; high byte of filename address
    jsr $ffbd       ; call SETNAM

    ; set start and end addresses in zero page for save
    ; 1024 = $0400; 1028 = $0404; $0404+1 = $0405
    lda #$00        ; <start_addr_low>
    sta $2a
    lda #$04        ; <start_addr_high>
    sta $2b

    ldx #$05        ; <end_addr_low>
    ldy #$04        ; <end_addr_high>

    lda #$2a
    jsr $ffd8       ; call SAVE kernal routine

    rts
```

LOADING

```
; first row, first column  --- 1024
; second row, first column --- 1064
; third row, first column  --- 1104

; cl65 -o loading -t c64 -I ~/cc65/include -L ~/cc65/lib -C c64-asm.cfg-MINE loading.s
; sys49152

    ; ***** ABC; filename; use ASCII
    lda #65
    sta 1024
    lda #66
    sta 1025
    lda #67
    sta 1026

    ; ***** loading
    lda #$01        ; logical file number 1
    ldx #$09        ; device 9 (disk drive)
    ldy #$00        ; 0: manually define address for memory storage
    jsr $ffba       ; call SETLFS

    lda #3          ; length of filename
    ; 1024 = $0400
    ldx #$00        ; low byte of filename address
    ldy #$04        ; high byte of filename address
    jsr $ffbd       ; call SETNAM

    lda #0          ; LOAD mode
    ; 1104 = $0450  ; third row, first column of the screen
    ldx #$50
    ldy #$04
    jsr $ffd5       ; call LOAD

    rts
```