
Notes:

- The main purpose of this week is to continually practice accessing files
 - Students are requested to submit the MIPS programs no later than May 05, 2021
-

Question 1. Following are instructions used to access a file (open/close/read/write)

```
# Sample MIPS program that writes to a new file.
# by Kenneth Vollmar and Pete Sanderson

.data
fout:    .asciiz "testout.txt"      # filename for output
msg1:    .asciiz "Before read: "
msg2:    .asciiz "After read: "
buffer_write: .asciiz "The quick brown fox jumps over the lazy dog.\n"
buffer_read:  .asciiz "-----\n"
.text
#####
# Open (for writing) a file that does not exist
li      $v0, 13      # system call for open file
la      $a0, fout     # output file name
li      $a1, 1       # Open for writing (flags are 0: read, 1: write)
li      $a2, 0       # mode is ignored
syscall      # open a file (file descriptor returned in $v0)
move    $s6, $v0     # save the file descriptor

#####
# Write to file just opened
li      $v0, 15      # system call for write to file
move    $a0, $s6     # file descriptor
la      $a1, buffer_write # address of buffer from which to write
li      $a2, 44      # hardcoded buffer length
syscall      # write to file

#####
# Close the file
li      $v0, 16      # system call for close file
move    $a0, $s6     # file descriptor to close
syscall      # close file
#####

#####
# Open (for reading) a file
li      $v0, 13      # system call for open file
```

```

la    $a0, fout      # output file name
li    $a1, 0          # Open for writing (flags are 0: read, 1: write)
li    $a2, 0          # mode is ignored
syscall              # open a file (file descriptor returned in $v0)
move  $s6, $v0        # save the file descriptor

```

```

#####
# Read from file
li    $v0, 14          # system call for read
move  $a0, $s6         # file descriptor
la    $a1, buffer_read # address of buffer read
li    $a2, 44          # hardcoded buffer length
syscall                # read file

```

1. Please write a program that allow users to input the number of elements in an array (call n); then ask users to input n elements of the array to store in memory; save the array to a file
2. Find and print the maximum elements in the file saved in the previous step

—————the end—————