# Convert this docx to PDF before submitting it

In doing the assignment when you search into the Armv8 datasheet for register def etc, be careful it has a mix of aarch64 and aarch32 things. Only aarch64 (64 bits) is relevant to us.

1. Modify boot.S, so that your kernel switches to EL0 (instead of switching to EL1).

**Note:** Some students may want to modify other file(s), which are not necessary. But it is fine you insist that. Just submit everything you changed. See below.

**Note:** you may find QEMU’s debug log useful, which traces the exception level.

a. (30) (QEMU users only) Attach a screenshot here showing that your code works. The screenshot may be messages printed from your kernel execution, or a qemu log.

b. (20) Briefly explain: what changes have you done? What register(s) do you have to touch? What values do you put in the register(s) and why?

c. (20) Upload your code a separate tarball. **Only include the file(s) you modified in the tarball**

|  |
| --- |
| How to pack a code tarball (for this & future submission)  File name: [computingid].tar.gz. e.g. lg8sp.tar.gz. All lower case. Use the command 'tar -czvf' to generate.  (Optional) A file README-cs.txt, which contains anything TA should know (e.g., any caveats of your code, extra build instructions)  Absolutely no any binaries (e.g. \*.elf, \*.o, \*.bin) or .git/ subdirectory in the tarball.  **# Sample command 1: to compress all source files under a directory**  **# Note: no trailing space after “\”**  cd YOURDIR  tar czvf ../[computingid].tar.gz \  --exclude='\*.o' \  --exclude='\*.d' \  --exclude='\*.bin' \  --exclude='\*.elf' \  --exclude='\*.img' \  --exclude='.git/\*' \  \*  **# Sample command 2: to compress specific files in a directory**  **# Note: no trailing space after “\”**  cd YOURDIR  tar czvf ../[computingid].tar.gz \  my1.c my1.h my2.c my2.h |

d. (20) Can you demonstrate that the kernel ACTUALLY reaches EL0? For instance, can you execute some instructions disallowed at EL0 before/after the switch, and reason about the results? Explain your choice and observation. Attach screenshot(s) if needed.

1. (30) After landing in EL0, can your kernel print out the current exception level?

If so, attach a screenshot showing the printout.

If not, explain why.

1. (10) What does an “eret” instruction do?

(10) Does an “eret” instruction must correspond to an earlier exception?

(10) What will happen if we execute an eret instruction at EL0?