**Goal of the first day: Preparation of experimental environment and confirmation of what needs to be done**

**1. Make a copy of ‘mtk-hina.tgz’ in ‘mtk’ folder to your own home folder, and develop it there.**

**=> The ‘mtk’ folder is formed and several files are developed under the ‘mtk’ folder.**

**2. Make a copy of the result (the simple OS) of the C experiment 1 to the mtk folder, and change the file name of the simple OS to ‘mon.s’.**

**Thereafter, works will be done in the mtk folder.**

**- Labeling the mon.s and adding the jmp instruction. (Text p.15, Sec.1.3.2)**

**- Implementation of ‘inbyte’ into ‘inchrw.s’ (Text p.17, Sec.1.4.1)**

**- Implementation of ‘outbyte’ into ‘outchr.s’ (Text p.18, Sec.1.4.2)**

**- Creation of ‘test1.c’**

**- Describe the operation test program by C language, in the main function.**

**- Make the test program looped, to be executed repeatedly.**

**- Prepare ‘test1.abs’ using ‘make test1’, and execute it on the actual machine. (Text p.18, Sec.1.4.3)**

**Notes**

**- Check the time points of starting and ending the hardware initialization in the mon.s.**

**- After moving to the starting position of hardware initialization using ‘jmp minitor begin’ and returning from the hardware initialization using ‘jmp start’, the system is in the state of the C library initialization until the execution of the C program’s main(). Therefore, the operation to make the function of the initialized hardware effective is to be incorporated before returning by the ‘jmp start’.**

**- The test program code described in the C program’s main() is to be prepared, not as the code that stops the system after running it one time, but as the code that runs the system repeatedly, to check whether it can be operated normally even when loads are imposed on the system. (The test code should be described in the infinite loop.)**

**\* The test code that stops the system after running it one time should not be evaluated.**

**- The description in Sec.1.4.2, “if the one character output doesn’t succeed in the PUTSTRING system call, it should be retried.” is also applied to Sec.1.4.1.**

**- The working area in the simple OS, ‘mon.s’, is prohibited to use. The reason may be naturally found, if the content implemented in the theme 2 is taken into consideration.**

**Items out of the scope of the experiment**

**-The assembly language’s pseudo-instruction, ‘.equ’, is related to the string substitution function, and doesn’t form the label with address such as a variable name and a function name. Therefore, it doesn’t function as expected, even if the reference to the label of external file is designated using ‘.extern’.**

**=> There is no choice but to redefine the ‘.equ’ one after another, or to apply the ‘.include’.**