Instead of writing our assembly code directly, we took a different approach. For computer systems I had written a compiler that compiles PHP code to the PP2 Assembly code. We have used a modified version of that compiler to create the code that now runs our machine.

For this reason we started by converting our Java Code to PHP. To make the conversion easy for ourselves we wrote the Java code in PHP style. In PHP all variables have a dollar sign at the start, so all our variables in Java we put a dollar sign in front of as well. As a result of how we wrote our code and because PHP and Java already look quite alike, our Java and PHP code is almost identical.

There are of course a few differences. Most notable are that instead of being able to put a variable into the memory directly, see line 149 of the java code, we have to first set a variable and then put that variable into the memory. This is because of limitations in the compiler. We also ran into the problem of having too many variables, therefore we had to unset variables in the PHP code, as you can see on line number 235 for example.

(NEXT SLIDE)

After converting the Java code we let the compiler do its job and got the assembly code. On the slide you can see the PHP code to the left and the Assembly code to the right. As you can see sometimes one line of PHP code takes up several lines of Assembly code. Also notable is that the compiler creates a “conditional” branch every time an If statement is used in the PHP code.

The assembly code we got, did not work. After some troubleshooting we found out the problem was not with the compiler, but with the PP2. It turned out the PP2 was not capable of having 2 motors on the same fuse. We had to modify our PHP code to take this into account, and the machine started working.

Now that we got our working code, we had to check if our machine satisfied our top priorities. After some tests we found that our reliability was xx% and our machine sorted all disks, even though it was significantly slowed down by not being able to use all its motors at the same time, in xx seconds. As Rolf said, we wanted at least xx% reliability and xx seconds so our machine satisfied our top priorities.

Anything unclear?