Work log Maturaproject

Moray Yesilgüller, Nick Gilgen and Daniel Huber

Week	Student and his work
17. March	***All***: On Wednesday we had our first meeting with our advisor, Mr. Palfinger, where we discussed the proceedings that will follow. We structured the approach of our project, deciding who makes what for the next few days. ***Daniel***: I wrote a boot loader for the OS so that we can start developing other programs. Because I have written multiple boot loaders in the past, this task was not that difficult. But for the first time, I have really tried to integrate error handling. I also read and studied the NASM docs, especially chapter NASM macros. On top of that, I started to write 3 print functions namely for Null-terminated Strings, hexadecimal numbers and signed decimal numbers. ***Nick***: I read more about the components of operating systems and the different types of operating systems. To start practicing I downloaded Oracle VM to install Manjaro, which is an operating system based on Arch Linux. ***Moray***: Simple pixel print macro written for graphics rendering, the program by printing pixels at an x, y coordinate. The program then proceeds to increment the coordinates to fill a rectangle in the desired size. Research in Intel Docs and other x86 assembly sources.
24. March	***All***: We made sure our work environments are all up to date and working correctly and adding the missing programs. We decided to use a makefile to simplify the compilation process. ***Daniel***: I finished the print functions and reworked the boot loader and tried to make the code more readable. I tried to implement an Interrupt Service Routine (ISR) for 0 Division Exceptions, but I encountered very weird bugs where the wrong effective address was written into the Interrupt Vector Table. I did not add it to my master branch for that reason. Wrote the makefile with targets all and run, the latter starting QEMU with the right parameters and running the OS. ***Nick***: I started reading the NASM docs and studied code of different open-source programs to deepen my understanding of assembly language. ***Moray***: Expanded on the pixel print macro by creating a 4-bit colored painting program with cursor to test aspects of the expanded macro. The program runs in VGA graphics mode and uses the aforementioned pixel print macro as a base for painting. At the time the program cannot save the drawn picture.
31. March	***All***: On 31. March, we met again with our advisor and discussed the terms of the contract. We then tried to fix the missing parts and reworked faulty statements. We are spending a lot of time reading Intel's manuals and reading NASM docs. ***Daniel***: Finally, I was able to find the bug in the interrupt handler. NASM calculated the wrong offset and loaded the wrong function pointer to the IVT. I had to debug using hex dump and looking through the machine code. I have written a macro that makes adding new IVT entries a lot more readable. The error was caused by compile time vs preprocessor time discrepancies. I started to write some documentation and texts about OSes and lower level computing. ***Nick***: I invested time in reading more about interrupts and studying program flow control. I want to continue my research and start creating small test programs next week. ***Moray***: Bug found in painting program it only allows a certain number of colors to be added. Looking for a fix of the bug.
7. April	***All***: We discussed the terms of the contract one last time and then we signed it. After careful consideration, we decided to use version control for working together more easily. Git seemed like a good option because it is very common. We spend a lot of time learning git to make sure we don’t mess up our repo. ***Daniel***: I created the GitHub repo and cleaned up the makefiles within the different subdirectories and I refurbished the directory structure. I read up on exception interrupts vs the 8259-chip generated interrupts. I was confused how both interrupts are hardware generated because I thought they would run the same ISR. Now I know that the 8259 can be reprogrammed to remap interrupt handlers. ***Nick***: I continued my research on assembly language and read more about memory addressing. I learned how to use NASM and QEMU. After that I started making small and simple programs to practice. I would like to make a very simple game to practice. In the holidays I will continue my research so that I can start making programs for our OS. ***Moray***: Started research on how to do string comparisons. Created new string comprison macro file that compares strings at given pointers. Uploaded string comparison macro and drawsquare macro files to GitHub repo. Both files require the firststage file in order to work, this file was also uploaded to GitHub repo.
10. April - 24. April	***All***: During the spring break we didn't work on our OS, but we each continued our research. ***Daniel***: I read a lot about the PIC and how to configure it. The PIC is a chip that allows hardware interrupts to be configured properly. I also looked for docs about the various timers and clocks on a PC system. I will try to configure the RealTimeClock to regularly create an interrupt just to learn how it works. I have not found a lot of time for coding because I was working during my holidays. ***Nick***: In the last two weeks I continued to read the NASM docs. ***Moray***: I started research into how the painting program can be made to work with video memory directly to make it faster and also add the possibility to add ASCII letters.

Week	Student and his work
28. April	***All***: Every student coded a programm and researched the instructions that are needed to archieve the programs goal. ***Daniel***: I read up a bit more on the PIC and I also wondered how a PC can be turned off. I found an article describing x86 Advanced Power Management which is used to control the energy consumption of hardware and can be used to turn off the pc or set it in a sleep state. ***Nick***: I read up on the CPUID instruction which allows software to report details from the processor. I started writing a program to get an overview on the processor's state. ***Moray***: I created a function that prints out all general purpose registers, segment registers and the flags in hexadecimal onto a video memory array. The programm still needs some programming to be done as the flags arent printed in manner that can be read but there are no registerlabels when printing it out to the screen yet.
5. May	***All***: We started using using video memory directly instead of using interrupts to print because it is a lot faster and because when an interrupt handler is running, no other interrupts are allowed to run. ***Daniel***: I wrote some test programs where I practiced and tinkered around using video memory, strings, processor specific instruction and pointers to get more confident and a bit better at writing assembly, mostly because in the coming weeks I plan on writing the filesystem and interrupt handlers and it still feels a bit overwhelming. Now I feel a lot more confident and I think that I have acquired a *best practices* feeling for assembly. I have implemented a macro that sets up the interrupt descriptor table. On top of that, I created a glossary and wrote a bit about the booting process. I also read a fair amount about VGA registers, keyboard drivers and a bit of SeaBIOS source code and altough it was very interesting, I got carried away a bit to much by it. ***Nick***: I spent more time on reading about processors since I started working on the CPUID program. The CPUID program now prints the CPU's manufacturer ID. I ran into some trouble when I tried printing another string before the manufacturer ID but was able to fix it after a while with Daniel's help. I also extended the code a bit so that it shows the model ID but I am not done yet. ***Moray***: Started working on a second iteration of the debug programm. This time the programm also works with buffers, labels for the printed registers have been added, flags are now properly printed out. Also looked into video memory.
12. May	***All***: All of us tried to fix a bug in printing string buffers. We have not ben able to fix it but we will try to debug it with hexdump and see if it was an an assembler error. ***Daniel***: I worked on debugging functions with Nick and Moray but we still have not found a solution. I also researched a bit about more about hardware generated interrupts and I also wrote a bit for the maturaarbeit chapters, specificaly the booting process and the inner workings of the CPU. ***Nick***: I continued working on the CPUID program. I tried using the CPUID instruction to get the model of the CPU but I invested more time into finding out why we cannot print string buffers. ***Moray***: I tried to write a trap interrupt handler. It would allow us to single step instructions during execution but the aforementioned string buffer bug hindered the process.
19. May	***All***: We tried to fix the video memory bug but decided to rather spend time researching topics and functions that we will add to our OS within the next few days. ***Daniel***: I took a bit of a break from coding and researched a lot about filesystems as well as their different types. This is because one of my virtual machines caught a filesystem error. I plan on creating a pseudo implementation of the FS in C and will translate it then manually to x86 assembly. Also, on Sunday evening I managed to get around the videomemory bug, wich was extremely relieving. I asume it had something to do with the assembler miscalculating the effective adress, similar to the bug I had in March with the interrupt handlers. Although I have not been able to identify the error completely, I am glad that it works now and we can proceed with our work. ***Nick***: I took a break from coding this week. I searched for a solution to our problem with printing buffers because it would help a great deal if it would be fixed. Luckily, Daniel found out how to print buffers. Now we should be able to continue programming at a steady pace. I also read about filesystems and the difference between ARM and x86 processors. ***Moray***: Because I was unable to get around the video memory bug, I spent my time learning a bit about filesystems. It turns out that they are really complex but I found it interesting nonetheless.
26. May	***All***: After fixing the bug that slowed us down each of us continued working on their programs. ***Daniel***: I started to write a filesystem concept in C. I still need to figure out an algorhithm that deals with keeping track of used and unused sectors. So far, the filesystem deals with superblock operations that are performed on the cached superblock and then written to the superblock on the disk. For the sector allocation algorithm I thought about using bitmaps where 0 refers to an empty sector and 1 to a full sector. In the C implementation I have not yet added such an algorhithm but I hope that I will be able to finish it within the next few days. Unfortunately, we have *a lot* of exams this week but we hope that we can finish our goals on time nonetheless. ***Nick***: I read more about caches because I wanted to include information on the cache size in the CPUID program. Unfortunately, I have put that on hold because after a while I realized that it is a lot of work. I have managed to print the cache size for newer Intel CPU models but it would take much more time to make it work for older CPUs. The method to get information on caches also differs from manufacturer to manufacturer. I might add this feature if I ever get bored or need something else to do. On a positive note, the CPUID now also prints the family ID and model of the CPU. It also prints the processors's brand string but I came across a problem with defining the length of the brand string's buffer. It is currently too long and the buffers that should be printed after the brand string are not being printed. ***Moray***: Created a keyboard scancode programm that prints out the hex data of a key when it is pressed. I have also found a bug in the stringcomparison programm but have not yet been able to fix it.

Week	Student and his work
2. June	***All***: We met up with our advisor Mr. Palfinger and found answers to questions that we had in regards to the intermediate submission. We worked a bit on the presentation slides and spent time coding the unfinished demo programs. ***Daniel***: I kept adding some features to the C implementation of the FS and realized that the hardest part is finding a reliable sector allocation algorithm. While I have an idea how to implement it, I've had trouble expressing the algorithm in code. I will keep implementing it in C and then translate every function to handcrafted x86 assembly. I also reworked the demo files because there were some bugs present. ***Nick***: I finished the CPUID program. The only thing left is to add a function that displays the system RAM, which is rather easy to do. I might add additional features to the CPUID program in the future. Other than that I created a wordlist for the hangman game, which I will work on during the next few days. Next week I will also work on the presentation on the intermediate status of our project and study for the many exams we still have. ***Moray***: I have written a clock, which reads the hardware device that keeps track of the seconds that passed since midnight.
9. June	***All***: We finished our intermediate goals and debugged our programs. we also worked on the presentation and decided that every one of us will talk a bit about their specific demo program at the presentation. ***Daniel***: I worked a bit more on the C implementation of the filesystem. I am not progressing as much as I would have liked to, because we currently have a lot of exams. Nonetheless, we worked on finishing the demoprograms and working together on Hangman was fun. ***Nick***: I made a simple text-based game called hangman. Moray helped me by writing a random function and Daniel wrote a function that uses the random function to load a random string from the wordlist into esi. Everything works like it is supposed to and there are no bugs so far. I also started on the program that prints the RAM size. It is more complicated than we thought. The interrupt we read about works only for DOS, so I had to find a different interrupt. I hope I can finish this program before the presentation on Wednesday. ***Moray***: Fixed a few bugs in previous programs and added the UTC labelling in the time program.
16. June	***All***: We held our presentation on Wednesday afternoon.***Daniel***: I worked on a hexdump function that allows us to view contents of memory at a specified location. It takes two arguments, namely *location* and *bytecount*. It prints *bytecount* many bytes to a buffer, which can later be printed to video memory. This is useful for detecting memory leaks and general debugging. ***Nick***: I looked into the bug that I had when implementing the ram detection. I am currently debugging it with help of Daniels hexdump. ***Moray***: I started looking into reading pixels from video memory.
23. June	***All***: We worked mostly individually but still exchanged ideas about new functionalities for the OS to work on. ***Daniel***: I started to gather some ideas on executables. I looked into ELF, the executable file format on most UNIXes. ELF is overengineered but also very futureproof. For FlamingOS we only need a relocatable executable file format. I will have a look at .COM files the coming week. They were used on DOS and might be more fitting for our OS. ***Nick***: The ram detection program now prints the details but I still have to sort the information and add commentary to the code. ***Moray***: Started research on how a shell works.
30. June - 4. August	***All***: We had our summer vacation but we rarely spend time together which made it hard to organize the pending tasks. ***Daniel***: Because I worked for 3 weeks during my holidays I took a major break from OS development. I only read some articles and the source code of the Linux process scheduler. ***Nick***: I have not worked on our project during the summer holidays. ***Moray***: I have not worked much on the project, I have only done research here and there.
11. August	***All***: We gathered links that could be interesting in the future. ***Daniel***: I researched a bit data structures used in filesystems because I was wondering if there was a more efficient bitarray scan method than the one I tested out and only works on registers. Turns out there is a x86 instruction called `bsf`, *bit scan forward* and it looks like this is exactly what I need. I also kept tinkering around with executables because I want to come up with a lightweight, simple executable file format. I looked into .COM and .exe (PE) and was surprised to find that every .exe program written for WindowsNT systems is built on top of a MS-DOS runnable version that just prints "This program cannot be run in DOS mode". I looked into UEFI boot and learned that it works by compiling the OS (or kernel or multistage kernel loader) to a PE file and telling the UEFI BIOS to load it. I found myself reading through Linux, UNIX and Windows lore and got to know the different mindsets that the developer communities have. Overall, I feel like my motivation has been restored and I an looking forward to the coming weeks. ***Nick***: I started reading more about drivers in general because I want to write a simple keyboard driver. ***Moray***: I started working on the shell and ran into a few problems, which I have already resolved.
18. August	***All***: We are still doing general research, not only about features we are currently working on but also topics such as virtual memory and system calls on modern systems. ***Daniel***: I started working on the bit scan functions that will be used in the filesystems block/sector allocator, specificaly it will keep track of empty and full sectors. Whenever a new file is written to disk, the sector allocator will scan through a bitmap where each sector's state (*EMPTY* | *IN USE*) is represented by a single bit. After allocating a sector to a file, the bitmap has to be updated again. ***Nick***: This week I spent more time reading about keyboard drivers. Besides that I refreshed my basic knowledge about C and C++ because of a project in IT. ***Moray***: I kept working on the shell, added more functionality to it and fixed some buggs. The shell had a problem in which it could not override previous characters which I have resolved.

Week	Student and his work
25. August	***All***: We are planing the presentation and figuring out how exactly we want to publish the booklet. We are also learning how to use the GNU debugger. ***Daniel***: Because I was having trouble implementing the filesystems sector allocator I decided to have a look at GDB, the GNU debugger. I remember reading once that it can be used with qemu. Because I am a GDB novice and know that Nick and Moray have never used it before I added the `make debug` targets to the Makefile, both in the demos directory and the kernel source directory. I also found a better way to implement the algorithm of the sector allocator using the `bt` (bit test) instruction. I am still debugging it. On top of that, I also continued to write more text for the booklet. ***Nick***: I have familiarised myself with the GNU debugger. I have also started working on the introduction of our presentation. ***Moray***: Uploaded the first iteration of the shell, at the current moment you can write and delete. The written string is saved into a buffer and can be reset with the enter button.
1. September	***All***: ***Daniel***: I had some trouble debugging the OS because I can't get GDB to dissasemble the instructions correctly. Qemu tells GDB that the CPU architecture is set to i386 (32 bit x86) and GDB interprets all the `JMP` instructions as a 32-bit instead of a 16-bit relative jump resulting in a shift of 2 bytes the following instructions are interpreted as garbage. Qemu will run just fine and GDB notices that the instruction pointer has been changed by the correct amount but the disassembled instructions are unreadable. I have created my own symbol table, a *txt*-file that maps the labels to memory addresses. This makes debugging easier because I can set breakpoints more easily without having to remember the memory addresses of every label. I found a gdb script on the internet that claims to fix the gdb 16-bit/32-bit disassembly issue but I have not been able to get it to work. It was written for a previous version. ***Nick***: I have decided that I will use either the scan code set 2 or 3 for the keyboard driver. The scan code sets define which bytes match which key on the keyboard. The third scan code set has access to more commands but is also more complicated while the scan code set 2 is the most commonly used scan code set. I plan on working more on the keyboard driver next week because I have no more exams after tuesday. ***Moray***: Fixed some more bugs in the shell it should now work as intended and starting to add the needed functionality like string compartison into it.
8. September	***All***: We started focusing a bit on the presentation and deciding which topics should be mentioned to the audience. ***Daniel***: I wrote more articles and focused a bit more on readability. I will have to rewrite the chapter Filesystems, especially the part explaining `.` and `..` as well as relative file paths. I think it is hard to understand for laymen. ***Nick***: I continued working on the keyboard driver. Unfortunately, I had a problem where my laptop shut down and I forgot to save the file. Now the program is back to its old state. My next step is to figure out how long the program has to wait for the keyboard input to arrive and be acknowledged. ***Moray***: Started adding a scrolling function to the shell. It should work by taking a byte from the next line and moving it to the previous one and repeating that for all letters.
15. September	***All***: We were working on the presentation and our individual demo projects and discussing topics that we should add to the chapters. ***Daniel***: I started working on the file chapter and am also working on the UI chapter. I also plan on writing a random number generator using a linear shift register. ***Nick***: The keyboard driver now has an identify function to detect ps/2 keyboards. ***Moray***: The scrolling function is now fully implemented. It still has a small bug that needs a workaround.
22. September	***All***: We had a lot of exams this week and did not find a lot of time for coding. We met up with our Advisors Mr. Palfinger and Mr. Bersier and got our questions answered. ***Daniel***: I wrote a bit more chapters and learned a few tricks in GDB to get better at debugging. I hope I can finish the sector allocator this week, because it has been ongoing for a long time. ***Nick***: I will start writing chapters on memory and processes. I have yet to test the keyboard driver. ***Moray***: Fixed more bugs in the shell. Will start on chapters now and upload the paint program as I have noticed that I did not upload it yet. Accidentaly deleted the paint program while trying to upload it.
29. September	***All***: We didn't work much on our code, but we also did some research and we continued writing the chapters. ***Daniel***: I wrote a random number generator using a LFSR. I have tested and it works but there is a bug where the seeds 0 and 1 result in an output 0 and set the seed to 0 aswell. After such an occurence, the LFSR becomes useless unless reseeded. ***Nick***: I read more about memory in detail, so that I can write chapters on memory and processes. ***Moray***: Started recovering the paint program via a binary file tht was left. The program still has problems in its disassembled form.
6. October - 27. October	***All***: We had our Maturareise that lasted two weeks. Even though we enjoyed the time, we worked a bit on our tasks and added new functionalities to our project. ***Daniel***: I debugged the sector allocator and I am confident that the filesystem will be usable. I plan on creating a flat filesystem first and upgrade it to hierarchical filesystem in the future, perhaps even after our Maturaarbeit is submitted. ***Nick***: I didn't do anything for the OS but I read a bit about memory access. ***Moray***: Started research into processes but not much as I was in vacation most of the time.

Week	Student and his work
3. November	***All***: We started working on the poster that we need to finish by november the 17th. We can now focus on the matura project during the next few weeks because we don't have any exams. ***Daniel***: I wrote a bit more for the chapters and learned a about logical block adresses (LBA). LBA is just sequental addressing for block devices such as a harddisk. It is much easier to convert from LBA to physical (CHS) addresses than organizing everything in CHS addressing. ***Nick***: I continued to work on the keyboard driver and rewrote the functions that access the I/O ports. ***Moray***: Started writing the processes chapter and also started working on recovering the paint programm.
10. November	***All***: ***Daniel***: I drew some graphs for the Maturapaper. They should help explaining some filesystem internals. I wrote a macro that helps mapping array indexes and struct member offsets to a memory addresses. This will be useful for accessing inode members inside the superblock. ***Nick***:Worked on keyboard driver and finally was able to access the I/O ports without issues. I also started writing memory sub-chapter. ***Moray***: Started working on the poster and kept working on the chapters.
17. November	***All***:We finished the poster and worked on our programs indivdually at the beginning of the week. At the end of the week we translated the chapters from markdown to LaTeX because of the limited formatting abilities in Markdown. Then we converted the document into a pdf. ***Daniel***: I have made significant progress in the filesystem. But I encountered a bug where the physical (Cylinder, Head, Sector) address to logical block address converter returns a faulty value. ***Nick***: Finished the keyboard driver and there are no bugs currently. I also started working on a text editor. ***Moray***: Fixed bugs on the shell, and worked on the journal and the scientific paper.
24. November	***All***: We finished the paper and split them into the research paper and a project paper. We then converted these into two pdfs and handed them in.