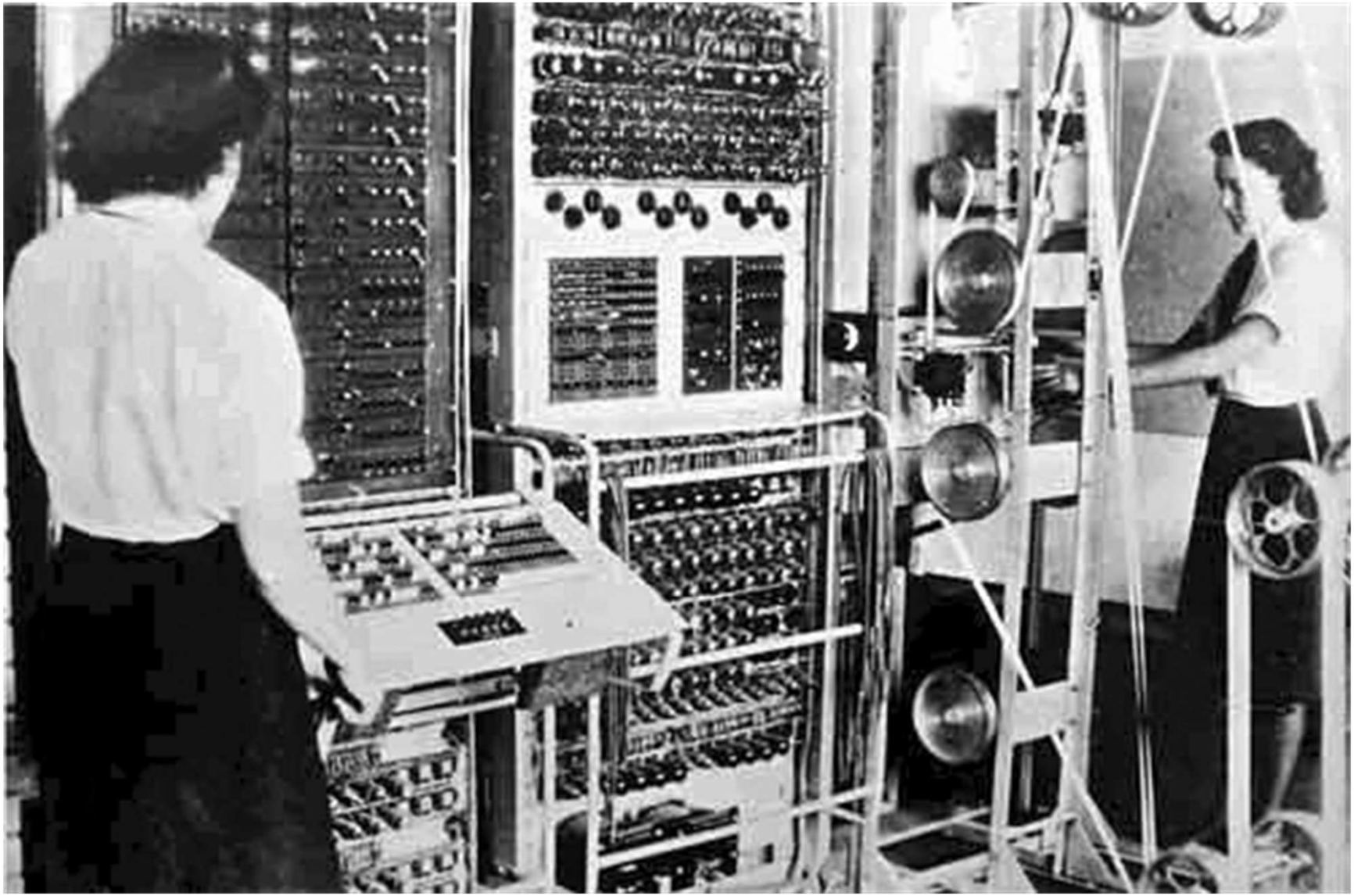
History of Operating Systems

The First Computers

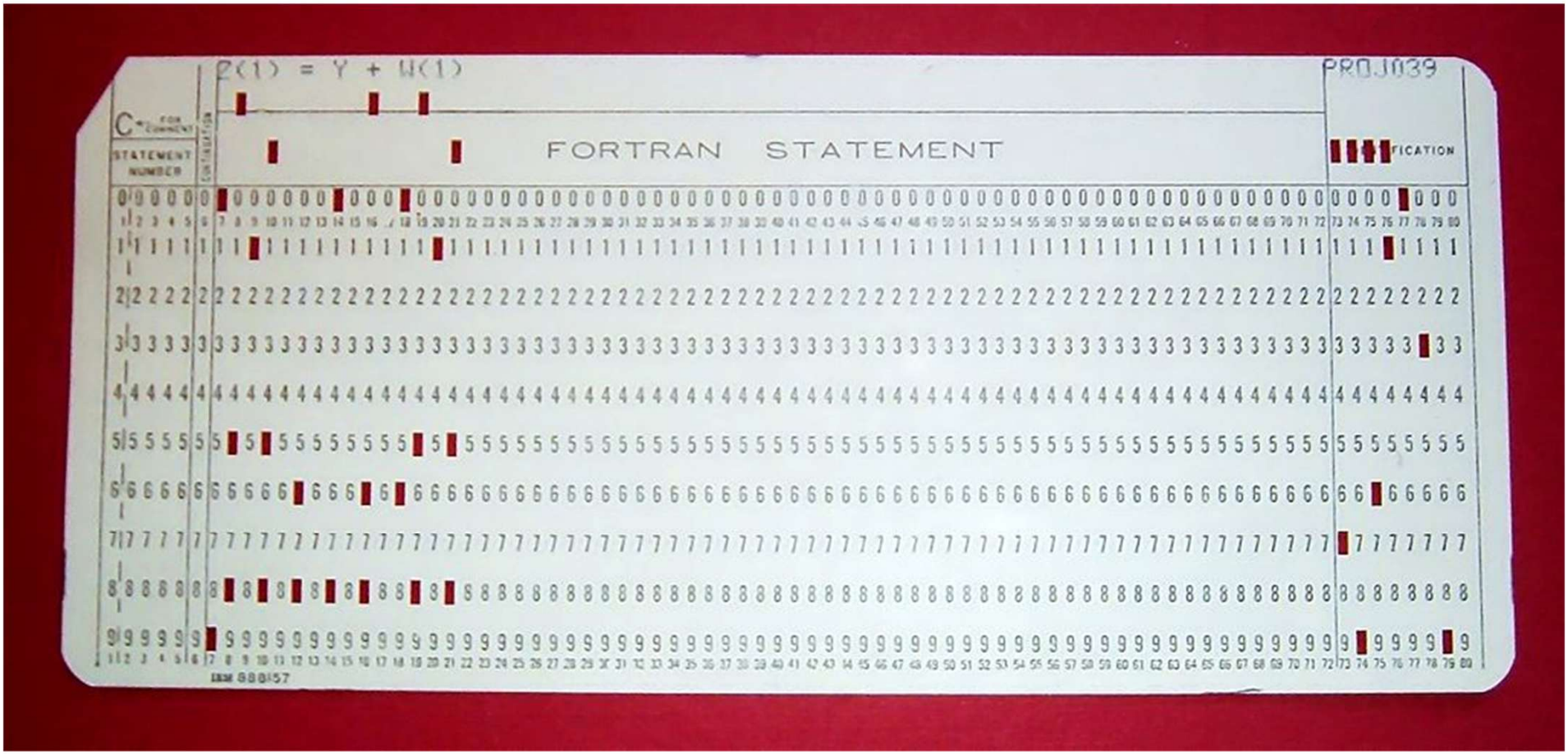
- Early machines (1940s to mid-1950s) had no Operating System

- The user interacted directly with the hardware
  - initial interfaces: console of switches (input) & lights (output)
  - later interfaces: punched cards, printers, etc.

Colussus



Punch Card


P(1) = Y + W(1)
PROJ039
FORTRAN STATEMENT
STATEMENT NUMBER
IBM 888157

The First OS(es)

- Created by General Motors in 1956 – called GM-NAA I/O
- To run a specific machine – the IBM 704 central computer
- Till about 1960s, different hardware vendors would write different OSes for each of their machines
- Then IBM started work on System/360 series of machines. The OS was called OS/360

Issues With First Computers

- Long setup time for a program to run
- Users accessed the system one at a time
- scheduling (what program is run next?) made by hand
- no sharing of libraries, drivers, . . .

Mainframes: Batch Systems

- The earliest Operating Systems were used in mainframes (1950s)

- These OSs were batch systems, which attempted to
  - eliminate the manual set-up of programs to be run
  - provide reusable code to access hardware (i.e. drivers)

Mainframes: Batch Systems

- Operating System was stored in main memory (was called a monitor)
- One job (program) loaded at a time from a punched card/tape reader into remaining memory
- Job control instructions told the Operating System what to do

Mainframes: Batch Systems

- These simple OSs were code to which one linked one's program (loaded as a whole into main memory) to be run

- Basically, the OS was just a run-time library

Issues with Mainframes

- Input/output (I/O) operations were very slow
- No computations were done while performing I/O
- This decreased CPU usage

IBM 701

Mainframes: Multiprogramming

- Idea: expand memory to hold two or more programs and switch among all of them (multitasking or multiprogramming)

- Multiprogramming systems were rendered possible by the first integrated circuits (IC) in the early 1960s

Mainframes: Multiprogramming

- Increase the processor utilization and attempt to optimize throughput (i.e., jobs completed per unit time)
- Degree of multiprogramming: number of jobs that can be managed at once by the OS
- Multiprogramming (aka multitasking) is the central theme of modern OSs

Mainframes: Multiprogramming

- Multiple runnable jobs loaded in memory at the same time
- Overlap I/O operations of a job with the computations of another
- benefit from I/O devices that can operate asynchronously (interrupts and direct memory access —DMA)

Mainframes: Timesharing

- Initially multiprogramming was still batch-based
  - turnaround time could be long for any particular job
  - no interactivity

- Idea: to have multiple users simultaneously using terminals, with the OS interleaving the execution of each user program in short quanta of computation

Timesharing systems

- based on time slicing (a.k.a. time multiplexing)
- each user feels like using the shared computer on his/her own
- challenge: to optimize response time
- it allows the users to view, edit, debug, and run their programs interactively

GE645 running Multics

Desktop Operating Systems (1980s)

- Very Large Scale of Integration (VLSI) circuits made it cheaper to manufacture complex hardware
- Hardware became cheaper
- Easier to have one computer per user than share mainframe

Desktop Operating Systems (1980s)

- usability facilitated by the introduction of graphical user interfaces (GUI)

- Idea: to maximize user convenience and responsiveness (apart from CPU & I/O use, such as in multiprogrammed & timesharing systems)

Mobile Operating Systems

- Handheld smartphones, tablets, etc.
- What is the functional difference between them and a "traditional" laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity

Parallel Operating System

- Idea: to run and manage parallel applications efficiently on tightly coupled parallel computers (multiprocessors)

- gives support for parallel applications composed of several time-consuming but separable subtasks

Parallel Operating System

- Provides primitives for assigning (scheduling) parallel subtasks to different processors
- Provides primitives for dividing a task into parallel subtasks, if possible
- Supports efficient communication between parallel activities
- Supports synchronisation of activities to coordinate data sharing

Distributed Operating System

- Idea: a common operating system shared by a network of loosely coupled independent computers
- Facilitates the sharing of resources located in different places (hardware and software)
- looks to its users like an ordinary centralised operating system

Distributed Operating System

- Supports communication between parts of a job, or between different jobs, across the network

- it allows for some parallelism, but speed is not the main goal

Real-Time Operating System

- Idea: to guarantee a response to physical events in a fixed interval of time
- used for specialised applications: subway systems, flight control, factories, power stations, etc.
- all activities scheduled in order to meet critical requirements

Real-Time Operating System

- performs operations within predetermined timeframes

- Soft real-time: implemented by all OSs in modern PCs to run multimedia applications

Practical

- In L023 (Sutherland Building) at 11:00 (right after this class)
- The practical will introduce shell scripting using bash. We will also use Javascript which is being done in the Web module [transferrable skill!!]
- This week, use the lab notes on Moodle, to set up a terminal for yourself, on your laptop