11.1 History of Operating Systems & 11.2 Processes

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-12-10 1:37 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 11.1 History of Operating Systems 11.2 Processes	Course: BSc Computer Science Class: How Computer Work [CM1030]-Lecture Date: December 10 2019

Essential Question:

What were early computers like and what are the processes that simulate to give the illusion of simultaneous execution in computers?

Questions/Cues:

- What is the name of the first fully-operational computer?
- What is thought of when mentioning early computers?
- What was the function/role of a punched card in terms of computers in the early days of computing?
- What was a time-sharing computer in the 70's like?
- What is Load Balancing?
- What is a process?
- What is thread?
- What is a time slice?
- What is an Interrupt?

Notes

- It was called EDSAC & it was built at University of Cambridge, designed by Maurice Wilkes
    - It was first "fully-operational" computer and in use, but not the first computer to be built, ACE built at University of Manchester Northern England was actually "working" first, but was a prototype designed by Alan Turing.
    - John Von Neumann, J Presper Eckert, and John Mauchly in US laid the technical foundations & came up with first plans for a comp
        - This work by used by Wilkes & Turing as basis for their machines
- Early comps consisted of no OS, screen or keyboard, just on/off switches
- Then came punched cards, which were used to enter programs into computers. To program a punch card, you would make holes in the card at the positions of the numbers you wanted to select. Those numbers represented machine instructions. The card would then be fed into special reader, then comp would interpret those machine instructions and run prog.
- In 1970's thing with a screen not a comp, but a terminal. All terminals did was input & output, & they were linked to the main computer (row of server racked box in A/C

room). Several people would be working on the same comp at same time using these terminals.
        - This was called a multi-processing or time-sharing comp
- Load Balancing = task of allocating apps to CPUs so as to most efficiently use computing resources.
    - Very important in multi-core processing
- Process = Is a coherent piece of software that is running on a comp & which can run at the same time as & independent of other processes.
    - can correspond to app or OS kernel, attached to app within
    - runs independently of other bits of code and its own memory
- Thread = Similar to process code, in that it runs independently as the same time as other bits of code, but it doesn't have its own memory.
- Time slice = After the process starts running & sends its code to CPU, it does stuff, but it only runs for a certain amount of time, after which it stops. This amount of time is called a time slice.
    - On single CPU, we are slicing up the amount of time we have on CPU between processes
    - When first process's time slice runs out, it gets an interrupt.
- Interrupt = A message sent to the process that says stop, "get off CPU, it's some else's turn"
    - The procedure of time slices & interrupts repeat for the remaining processes and then goes back to the original or first process. But processes don't always happen in order, sometimes processes are simply high-priority or that a process needs time to load a resource/data from somewhere and cannot execute until then; other processes would run and do their bit & once data is obtained it can start again instead of everything stopping and waiting.

Summary

In this week, we learned about the history of early computing & the processes behind the scenes that execute in a timely manner to provide fast & easy access to a user using a computer.