January 5, 1971

Research aspects of Multics

Problem: no usable model of presented load for a general-purpose computer utility

Consequence: simulations don't work; must build and test ideas on live users

Technique:
- propose a system with certain properties
- construct a prototype, noting engineering compromises which provide insight
- try it out on live user community; note usability under realistic conditions
- iterate, trying to discover simplifications and underlying structure

Limitations:
- Cannot propose a change so radical that users cannot understand it. (M.I.T. users can swallow fairly radical changes.)
- must keep an economically viable solution in order to attract users
- must control rate of change and avoid experiments destructive of user confidence
- user community may not be representative of broad enough class of users. (M.I.T. is quite broad)

Virtues: Assured contact with reality

Future research directions

Overall research problem being tackled: we hope to turn the fabrication of large scale information-processing systems into routine engineering development projects. This objective requires learning more about the conceptual understructure of such systems, in contrast with a more limited objective of inventing ingenious techniques of fabrication.

Specific topics:

Really large files

now $10^{10}$ bits. (IBM 3330).

need to understand $10^{12}$ (100 times as much) protection (access control/privacy)
reliability
performance
System Interconnection (Networks)

privacy
protocol
sharing at a distance (interlock/ownership)
bandwidth prediction

Modeling of the virtual memory interface

presented load models; models of sharing
predictive page movement schemes
constructive prediction of memory performance from
device characteristics and presented load model

Other areas

simplification of mechanisms everywhere
fault/interrupt handling; protection schemes
mutually suspicious programs
system implementation language studies
scale up to 1000 users - what happens?
information retrieval in the virtual memory environment