

Notes for talk

8/12/69

History of Multics, reasons for doing, clear CTSS problems  
length of project, current status. (brief)

Follow hardware lines; touch on hardware

DSU-10 → DSU-270; capacity a year or so ahead  
plan (tentative) to bring in 2314-style (DSU-120)  
to get out of DSU-10 mode, if necessary.

Development machine currently used for testing; verification  
currently underway as to its future status after 12/31.

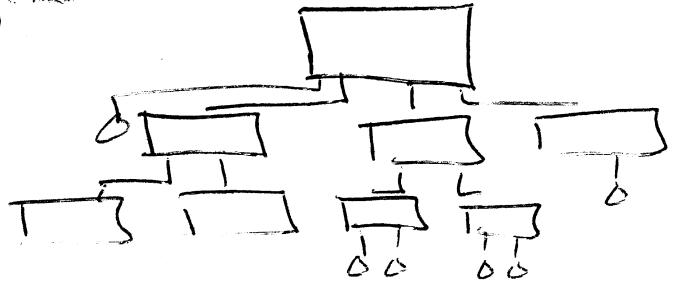
Glac equipment for M37 TTY, 1050/2241, AKDS  
possible addition: TN 300, M35/33. (Dura Corp.)  
(only thing going for the 33 is that everyone seems to know it)

### ~~System Organization~~

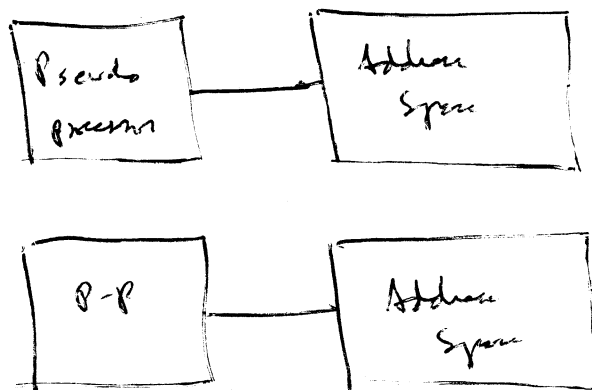
Hardware can grow to 3-4 times present capacity; organization  
allows for modules to be plugged in one at a time.

# System Organization

## 1. Hierarchically Organized File System



## 2. Multi-process environment



- Address space can overlap.
- One gets at files by mapping them into his own address space.
- Supervisor is a set of processes appearing in your address space; it is shared.
- Interprocess communication (control) is provided.

3. Reliability is not quite up to CTSS, but it is still very usable. A "stable" system will have 1-2 crashes/day (probably software bugs). Occasional hardware instability will keep system down for a day. New software systems sometimes are buggy. (Vintore: many bugs only kill on us.)

<sup>4/11</sup>  
Plans.

IPC has committed itself to run Multics for at least 3 years; GE is solidly behind it with possible new hardware, etc. in future.

CSK Gray will continue to upgrade it; not a on-site expert in system.

ARPA Net, Display Support, Upgrade hardware, new functions (e.g., absence)

### Documentation

~~User's Manual~~

Programmer's Manual (In Draft) (Good Draft about Oct, 1)

Condensed Guide

Subsystem Writers Guide (Available Drafts 1-9; 8 coming, publication this year)

Papers (5 coming this fall)

Internal - MSA, etc.

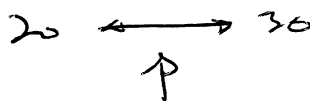
- BC/I : Mission : - I/O, File system  
 (One Multics calls to achieve  
 made more sophisticated versions)  
 - Sealed fixed point (Sandy)  
 - Turing

Most everything else is there, implemented, and working.  
 e.g., structure, pointers, ~~...~~, automatic  
 type conversion, string functions (quite  
 good implementation) etc.

This is our 2<sup>nd</sup> computer, and it has fitted a lot —  
 It produces very good code.

Performance, etc.

Depends on class of user.  
 Today { PDP-8 simulator interface that's a light user.  
 Daily usage 20 very heavy users.



actual # depends on volume.

I expect to see the 30 end climb to the 50-60  
 area during next year. 20 might go to 25 at most, since  
 limitation is outside of replacement; speed of bits.

Current scheduler is biased toward heavy users, and  
~~is profligate in CPU usage.~~

gives every user like it was cleaner. Result: miserable

response on small jobs. This will be fixed by OS.

### Disadvantages

1. GE is a different path than IBM.

Community of users consists of

GE 625, 635 users

GE 605 users

Future customers of 645

(Several have shown interest in past, but

have adopted a "wait for the software" attitude.)

PL/I communication possible  $\in$  IBM

ASCII char set maps into EBCDIC with some care. (Data)

FORTRAN communication possible

2. System is new; there are lots of rough edges to be polished  
and heavy engineering to be put through before we  
are satisfied. (But system is workable)

One process;

Protection rings;

Ring-0 contains supervisor

Ring-1 contains command language runtime, library.

User can write additional rings if he wants.

All commands are in distinct registers from user programs.

Type "x"; ~~di~~ <sup>di</sup> ~~variable~~ <sup>variable</sup> and directory,

then library, for

something named x, then

write it in, then call it.

cell "x", taken ~~for some~~.

5 hours / see keyboard

Language:

FORTRAN IV Complete, working, has been used only a small amount, will probably show up with minor bugs. No NAMELIST; some simple, almost

~~simple~~ ~~free~~ ~~form~~ ~~I/O~~ will be provided in to supplement the usual formatted I/O.