THE GRUNT PRESS

(and other wrestling holds)

ISSUE NO. 9

DATE November 26, 1982

COMPANY CONFIDENTIAL

Under new Management, but still full of the goodies that you know and love.

All of you Roosevelt subscribers get the morning edition. (Yawn)

/ Ked.

From the cave of Leo Binkowski:

As most of you at Roosevelt may know by now, I am the new Editor of the GRUNT PRESS. My experience involves only that which comes from reading comics and motorcycle magazines. This makes me perfect for the role of editor for the GP.

For those of you who don't know me, remember the infamous Halloween party at Chuck's. I was the third hoser, (or the first, depending on which way you look at it), that had the opener tied to his parka. (Talk about weight watchers, I lost more than a couple pounds in my parka that night.)

Life here in the Games room dungeon consists mostly of long, boring nights watching a fuzzy TV spit out the results of your game, listening to Greg mumble about his program, and asking Ken a simple question, only to be greeted by a loud, "WHAT???", as he removes the headphones of his walkman to hear your question for another time.

It's my opinion that Todd has grown roots in his chair and therefore can't leave it to go home. As I stroll by with my Hot chocolate and say,"How's DEMOTRONS coming". I don't get any voice reaction. All I know that when he bears his teeth, It's time to leave.

I have to mention what a super job John Shortt is doing to make life easier for all of us here in the hole. Not only is he around most of the day to answer our questions, he usually stays into the night for us part-timers seeking opinions and advice. Keep up the good work, John.

Now for the work I'm doing (No comment from you, Tim). Ken and I are working on the home NABU version of roulette which uses the 'double-the-patterns-available-on-one-screen' trick. Ken will be doing the 3-D spinning wheel, while I'll be doing the payoff, betting, and housekeeping routines.

One final comment, Santa will probably be watching all those mischievous little boys who play with other people's VAX accounts.

From the desk of : Trevor Pearce

First off, I'd like to extend the best of luck to the new LED (thats Leo the Ed). I would like to thank everyone for their support in getting the GP rolling and I hope you will continue to supply the good info and sharp wit which is the heart of the GP.

Secondly, I'd like to recommend that any insomniacs try reading 'The Soul of a New Machine' by Tracy Kidder ($3.95 paperback). It is the sometimes interesting coverage of the development of a new computer from the paper design through 'getting it out the door' and all of the associated highs and lows along the way. The book is not too technical and minicomputer fans will chuckle at some of the analogies used to describe the operation of the various hardware and software components. Non-technical people can easily skip the machine descriptions and not miss too much. The interesting parts deal with the human and corporate aspects (backstabbing and mushroom management) which could easily be paralleled to some Ottawa high tech outfits. It may be a surprise to some but even the almighty Data General resorted to child labour and sweatshop techniques to try to put a product out on an absolutely absurd schedule. I highly recommend this book as pure fiction for some, a slap of reality for others, and unreadable for a few.

Now that my full-time GP requirements have eased up, I can get down to some real work. Over the next month or so I will be putting together a report dealing with the Application Development Environment (ADE). The ADE is concerned with any and all activities, processes, tools, machines, documents, black holes and sh__ luck involved with getting an application from its initial conception through to being a product on the data wheel. Hopefully, I will come up with a good description of what currently exists, some possible future ADEs and some possible plans for progressing towards a more ideal ADE. (The last thing we want to get stuck with is a lemon-ADE! Arr, Arr)

Notes:

Eric is making great progress on CABLESIM and should be able to wrap up most of the 'version 1' features soon. The big problem is getting a board to work with. The 'powers that be' were so impressed with John Allen's quack and ducky version that they sent it to Toronto along with Eric's board!

John's System Test Software is humming nicely and has been getting rave reviews at Almonte. John is currently conducting a languages study and will appreciate any help you can provide.

Dave Allan has hung up his diskettes and moved on. Anyone who uses his DEBUGger will agree he has done a really fine job.

From The Messy Desk of Bill Bourne:

1. The IOS for the NABU P.C. is now officially released! There is a copy in the program library which we will keep current. Also, if you find any bugs in the IOS, or have suggested improvements please fill out a software change request form and we will try to get the problem fixed up as soon as possible.

2. Trevor Pearce and John Harley will soon be breathing down your backs looking for your comments on programming labguages and Applications Development Systems. This is your chance to complain about what we got now, make suggestions about what improvemnts we can make, and generally dream out loud about what utopia should be like. The intention (now that we've met the Nov. 15th deadline and have the opportunity to get sensible) is to gather information from all the Content Implementors and produce two documents based on that information. One will outline the requirements for the Applications Development Environment (ADE), the other will outline requirements for programming languages.

So its a chance to have your complaints listened to and acted on! Coming to a basement near you! Don't miss it!

To the GRUNT Press:
From Dave Allan's last stand:

Before I leave (possibly to return if the Lord NABU works in the strange and mysterious ways I suspect he does!), I would like to leave these words of wisdom to those shafted into using PLMX:

1) Use another language

If that fails,

2) Ensure CASE statement arguments are of type ADDRESS or your program will also work in strange and mysterious ways. Type BYTE will work till about Case 7.

3) Do not use type BYTE fields in STRUCTUREs as DO indicies, the generated code will use the index as an ADDRESS type variable.

4) Make sure STRUCTUREs and Arrays with DATA or INITIAL statements are dimensioned correctly. The compiler does not check that if your initialization overflows. (The remainder of the initialization appears to be appended to the WSECT for subsequent WOM burning).

Lately I've been beating the 1100/HCC debugger with my hammer in an effort to make it small enough to link. The fact that DEBUG takes up 50 odd K or so and ZSID fits into 8K tells me somthing about both my programming style and PLMX. (It also means I did a lot of beating). Ultimately I extended Arthur Hams PLM optimizer and I understand that it has been incorporated into the library. I would caution initial users to ignore the "GOT TO E,A/D,0 OPTIMIZATION" message until it goes away. I'm like IBM, I release immature software. For your patience with the "GOT TO..." messages you get not 5, not 10 but a 15 to 25 percent code size reduction. Whats more, it's a clear example that space and time are not always traded off. Do not optimize your mainline module unless you edit in a global at the desired transfer address location as the origional will be removed.

The final (mature) version of the DEBUGGER is now available with SOME of the features that everyone (especially Warren, like DAILY!) indicated to me would be very desireable to have. The big news is an Assembler, Symbol table loading optional, and Enable/Disable interrupts. The new version is 2.2. If you are using an earlier version trade it in as there are several identified bugs in earlier releases which on the odd chance you haven't already encountered yet. The identified bugs are:

1) X SP=nnnn does not work.

2) Set memory does not make breakpoints transparent.

3) Expressions in the Dump length field will cause the disassemble option not to be recognized.

Plus a few others which don't come to mind. When you pick up your copy of 2.2 read the documentation as the restrictions on usage are much more completely documented than in previous releases.

If the debugger does somthing childish, complain to AIDs and they will probably complain to me and perhaps I will be in a position to do somthing. I suspect development of the debugger is at an end as any major improvements will take a disproportionate effort.

From the hidden reaches of Warren Belkin's mind...

Hi, if your a programmer, welcome. If your a commie terrorist, put this paper down right now. If you're a WSF (no names mentioned), take a hike.

Well all you fellow grunts, I am sort of back to the land of the living, although I am starting to doubt that the total lack of feeling in my body extremeties is due to overexposure to VDT's! I am a very (or at least somewhat) happy fellow right now. After much effort, several sessions of ocult divination, and much application of the Blessed Brew, my menu is on the air and operational. I actually made it with an entire 24 hours to spare before deadline (which was used to consume several healing liquid remedies)!!! Actually, talking about remedies, some of you may have been aware of my physical state two weeks ago, which was due to medicine received for Strep throat. Boy, I didn't touch the ground for days. Anyway, I am now finishing off the trivial paperwork for the menu, things like coding specs etc.. I should be free of this burden within several days (he says breathing sighs of relief). My next project is somewhat uncertain. I may make practical use of my latest academic achievement, that of advanced grade "c" basket weaving. At least this would provide cages for certain GP editors (that means Great Programmers-- Led.)and various funny sounding animals downstairs (one keeps going PONG, PONG, PONG...). Also, I would like to report a first in artificial intelligence. It seems the VAX has begun to think for itself, and has actually sent a certain unamed person a message (hint: his first name is an astrological sign). Anyway, my congratulations to those who worked so hard to get games up for the 15th, and to those who didn't, take off eh... So now that I am finished my project, I will slowly return to being only a slightly off base person, instead of a crazy. I appologize to all those I have insulted, sweared at, and generally been grumpy at the last month, but I was under some pretty severe pressure, and was doing the best I could. Within another week, you should be able to insult me without getting hit... Well enough of this rambling on, I bid one and all adieu. 73's, and to some 88's. Warren.

FROM THE SOMEWHAT SEMI-FUNCTIONAL PROGRAM LIBRARY:

Yes, you guessed it, we are still CORVUSless at this point and it looks like we will be for a little while to come yet. The service people at CI sent the CORVUS off to CORVUS Systems to be checked out. I hope it'll be back by the end of next week or early the following week, but I'm not holding my breath. So until then you'll be faced with a bit of inconvenience. It just may take me a little longer to find what you want, but I don't expect any major problems. If there are any routines or anything that you want to submit to the library, we're open for business and more than happy to help. (That's the royal 'we', you understand).

Down to the nitty-gritty (a.k.a. what's new in the library):

1) There is a new and improved (?) Software Change Request (SCR) form. This is not radically different from the old one; the only alterations are to clearly indicate the part you are to fill in, and to distinguish the section that is for library use only. When you have a change you'd like to see made to a piece of software you can pick up an SCR from the library.

2) A new release, courtesy of Todd McNaught: FILFIXER.COM, found at the CORVUS site, contains the following utilities.

1-Convert a document file to a non-document file (All ^ but TAB and SP omitted)

2-Convert a BASCOM print-file to a source file

3-Convert a M80 print-file to a source file

4-DEDOC a file yielding only the source code file

5-DEDOC a file yielding only the comment file

6-DEDOC a file yielding both the source code and comment files

7-Shift each line in file right to a specified column

8-Shift each line in file left x number of positions

9-Delete all characters in PRECEDING a specified string

10-Delete all characters in AFTER and INCLUDING a specified string

11-Delete all occurrences of a specified string

12-Delete and substitute all occurrences of a string with another string

14-Concatenate a string to the end of all lines at a specific column

15-Optimize a PLM MACRO file

16-Sort a symbol table into ascending numerical or alphabetical order

17-Get a disk directory (Upper or lower case filenames displayed)

18-Rename a disk file (Upper or lower case filename accepted)

19-Delete a disk file (Upper or lower case filename accepted)

3) The RS-232 debugger files DEBUG.COM and DEBUG.DOC are now officially released and are on the HCC DEBUG disk at the CORVUS site.

4) Changes to the PL/M optimizer : the files OPTIMIZE.BAS and OPTIMIZE.DOC are on the PL/M diskette at the CORVUS site.

Everything else is pretty much as usual down here. When the CORVUS reappears, I will continue the reorganization and updating of the Library.

A message to a certain 'programmer' - I keep telling you when my birthday is, but you just don't pay attention!!!!

Kathleen

FROM THE DESK OF GREG ADAMS

OR SHOULD I SAY SEMI DESK SINCE WHO ELSE BUT LAURA SCHENNING MANAGES TO OCCUPY IT EVERY TIME I APPEAR. OH WELL THAT'S LIFE.

TIME TO SAY SOMETHING IMPORTANT.....
THE NEW HIGH SCORE ON..

```
ZZZZZZZ      A A      X     X    X     X    OOOOO    N     N
     Z     A     A     X   X      X   X    O     O   N N   N
   Z       AAAAAAA       X          X      O     O   N  N  N
 Z         A     A     X   X      X   X    O     O   N   N N
ZZZZZZZ    A     A    X     X    X     X    OOOOO    N     N

                      IIIIIII     SSSSSS
                         I       S
                         I        SSSSSS
                         I              S
                      IIIIIII    SSSSSSS
```

238 800

+

3
SHIPS

REASON FOR 3 SHIPS: BOREDOM..GAME TO EASY..

ALSO I WOULD LIKE TO KNOW IF ANYONE HAS BEATEN MY NEW HIGH SCORE ON

VENTURE

IT IS 1,000,000 WHICH I ESTABLISHED AT HOME..

+

5 MEN (SAME REASON AS BEFORE)

YES LEO THERE ARE WITNESSES TO THIS SCORE.

NOW ONTO OTHER LESS IMPORTANT MATTERS.. NOW THAT WE HAVE A NEW EDITOR (LEO B.) THE GRUNT PRESS WILL BE ON TIME TO THE SECOND (NOT SAYING THAT IT NEVER WAS).

GOOD LUCK LEO HAR HAR HAR HAR... IF YOU EDIT THAT LAST ONE OUT THE PHOTO COPIER WILL BE JUST.

AS SOME OF YOU KNOW I AM CURRENTLY WORKING ON PONG. ALL IS WELL SO FAR . I WOULD LIKE TO THANK ARTHUR HAM FOR ALL THE HELP HE HAS GIVEN ME THANKS ARTHUR.

ONE LAST NOTE ...DOES ANYBODY OUT THERE KNOW KATHLEEN PICKARD'S BIRTHDAY (NOT YEAR AND MONTH ..DATE).

IF SO PLEASE RESPONDE` EMMEDIATE` MON(NNE)..

BYE FOLKS.................

OH INCASE YOUR LOKKING FOR SPELLING ERRORS LEO..HERES ONE JUST TO KEEP YOU HAPPY CINA DORR IS THE DLMDUS

FROM the perspective of Arthur Ham's desk:

```
'______________'
|-_          _-|
|  -_        | |
|    -_   ___| |
|     -__-|  | |
|     _L_ |  | |
|   _-/  -|  | |
| _-/    '---| |
|_-/        \-_|
'--------------'
```

I know there's some cheese in here some where.

"He used to spend his Sunday afternoons on the fifty yard line. Now he spends them in Hyper Space."

A Style Manual for Authors of Software

Kevin Mackey and Twila Slesnick

Although the authors wrote the following guidelines with educational applications in mind, their suggestions are equally appropriate for business, recreational and general programs.

In the wake of rapid developments in computer technology, creators of software are scrambling to fill the need for programs to be used with each new computer. There is now an abundance of software. Only a small percentage of it is educational and the quality of these programs varies dramatically.

This is largely because educational programs have been written by hobbyists, educators, publishers, software companies, and computer companies. There is no consistent format, standard, or quality. This can make the programs difficult to use and understand.

The following is a list of program standards established at the Lawrence Hall of Science. This list can serve as a guide for people who are writing or modifying programs for use in informal as well as formal settings. For the most part, these standards are computer independent, and therefore could apply to programs written for any computer.

The Beginning of the Program

1. Have remark (REM) statements at the beginning of the program to identify and describe it. External documentation may be lost, so it is necessary that a program contain its own. The REM statements should include the following information:

a. Program name.

b. One line description of the program.

c. Computer for which the program is written. (Note the model or ROM version, when appropriate.)

d. Minimum amount of memory (in K bytes) required to run the program. (Include memory needed for variables.)

e. Language or language version (e.g., Integer or Applesoft).

f. Special peripherals needed (joysticks, paddles, printer, disk).

g. Files used (e.g., on Apple, which fonts, machine language subroutines, etc., are read in from files).

h. Author(s) or author's organization.

i. Address of author or author's organization.

j. Date written.

k. Copyright statement or statement of permission to copy.

l. Modifications made.

m. Author of modification.

n. Date of last modification.

2. Set the mode of the computer as needed. For example, if you are going to use graphics and uppercase letters on the PET, the computer must be set for that mode. On the Apple, you will need to set up the text page if you are going to print text. Do not assume the computer is already set correctly for the program.

3. Clear the screen so that remnants of other programs are not visible when the user begins.

4. Print the name of the program along with a greeting (e.g., "Welcome to the game of Caves"). This will allow the user to verify which program is in the computer.

Use diagrams for explanations whenever possible.

5. Ask the user if he would like to see the instructions. If the user is familiar with the program, he may want to skip the instructions.

Instructions

6. Double space the text so that it is easier to read.

7. Limit instructions to three screen pages in length.

8. Use diagrams for explanations whenever possible.

9. Have a character (e.g., "B" for back) that allows the user to go back to the previous page of instructions.

10. After the instructions are printed, ask the user if he wants to use the program (e.g., "Do you want to play?"). After seeing the instructions he may not want to try it. If you do not want the user to have a choice, omit this step.

11. Include reminders throughout the program about how to play. For example, if the user is to enter a coordinate, the program might remind him to enter two numbers separated by a comma.

Kevin Mackey and Twila Slesnick, University of California, Lawrence Hall of Science, Berkeley, CA 94720.

Input

12. Use input routines rather than input statements. The Basic input statement has several pitfalls which can be avoided. Your routine should:

a. Clear the keyboard buffer before receiving input. Previously typed characters may be mistakes, so they should be ignored.

b. Prevent the program from ending if the user presses the Return key without having input any information.

c. Prevent the user from deleting too many characters when correcting his errors (i.e., users should not be able to delete the text of the program).

d. Screen out characters that would clear the screen, home the cursor, or move the cursor if the user typed them.

e. Limit the number of characters the user may input. For example, if you want to accept a maximum of nine characters, the user should not be able to enter more than nine characters.

f. Have the cursor move to the left and erase one character at a time when the user is deleting errors. Characters printed to the right of the cursor should not be dragged to the left as the cursor moves.

13. Have the program wait for the user to press Return after his input, rather than going on with the program as soon as any key is pressed.

Allow the user to watch his own progress through an activity.

14. Check the input for errors. If errors are made, print an explanation of the error and then ask for the input again. For example, if a user enters a letter instead of a number, an error message might say, "Remember, you can only use numbers."

15. Allow enough room after a question (e.g., "What is your name?") for the input to be typed without wrapping around to the next line.

16. For Yes/No questions, check only the first letter that is entered. If it is not "Y" or "N," ask the user to type "YES" or "NO" and then repeat the original question. This allows the user to enter either the whole word or just the first letter.

17. Check for errors the program might make as a result of user input (e.g., dividing by zero, string overflow, etc.). Give an error message to the user at the time of input.

Throughout the Program

18. Use a page format when printing to the screen instead of scrolling or using timed displays to allow people to read at their own rates. This format should:

a. First clear the screen.

b. Print at most one screen of text. (Be careful not to print so much that the text scrolls.)

c. Save the last line of each page for printing error messages or prompting messages such as, "Please press the Space bar to continue."

d. Go to the next page when the Space key is pressed.

e. Scroll only when necessary. For instance, when writing a story, the user may want to see not only the last line he wrote, but several preceding lines as well.

19. Use the Space bar for prompting (having the user tell the computer when he is ready to continue). This key is large, and rarely used for anything else. Do not use timed displays.

20. Begin and end words on the same line. Do not hyphenate them or have them begin on one line and end on another (wraparound).

21. Allow enough room in PRINT statements for printing values of variables without disturbing the output format (i.e., do not cause wraparound).

22. Allow the user to return to the instructions, stop the activity, or restart the program at intervals throughout the program.

23. When the program is going to take some time to make calculations or fill an array, print a message on the screen asking the user to wait.

24. Allow the user to watch his own progress through an activity when appropriate. In guessing games, for example, the user should be able to see his previous guesses to help plan the next guess.

The End of the Program

25. After the activity, ask the user if he wants to try the program again. Allow him to change the conditions (e.g., difficulty level, time limit) if he wants.

26. When the user finishes or interrupts the activity, have the program:

a. Clear the screen.

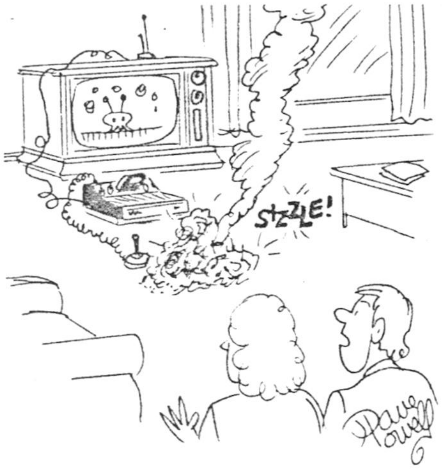
b. Comment on user performance, if appropriate.

c. End, leaving comments on the screen and placing the cursor on a clear line so that a command can be entered; or

Ask the user to press the Space bar and then run a menu program which lists the names of the available programs.

The above standards reflect an attempt to make programs self-explanatory, easy to understand and use, and relatively free of bugs. We welcome suggested additions to this list.

Those readers who are interested in other articles on program standards might look at Berentson and Ahl's "Sesame Place Style Manual" in *The Computing Teacher*, September 1979; and "Microcomputer Software Development: New Strategies for a New Technology" by Kehrberg in *AEDS Journal*, Fall 1971. □



"My God!--I think the Invaders finally wiped out Andy!..."

From the desk of John Harley:

Hi Folks! Bill Bourne and Trevor thought that it would be a good idea if someone looked into what programming languages we should have. To date, a serious study hasn't been made of the subject and there has been some occasional dissatisfaction voiced about the existing programming languages. So I've taken on the dubious pleasure of doing the study. So far they've been doing the planning and I've been doing the work 'cause I'm just a co-op student and don't know enough of what's going on. Actually, I might have that backwards, but at any rate, we are taking a serious look at available programminng languages and what it is that you want from them.

Now we come to the difficult part. This is where I get down on my hands and knees and beg you to fill out a questionnaire. I wouldn't ask you to do this normally, but Bill says we have to consider what it is that you people want from a programming language.

1 YOUR NAME

Your name? ________________ (Don't worry, they get easier)

2 PREVIOUS PROGRAMMING EXPERIENCE

What types of programming have you done, in what languages and on what machines?

What languages have you tried at NABU ?

3 FAVORITE LANGUAGES

What is your all time favourite language on any computer?

why?

What is your favourite programming language at NABU :

PLMX __ C __ BASIC __ FORTRAN __ PASCAL __
STRUCTURED MACRO __ Z80 ASSEMBLER __

Why?

4 COMPLAINTS

If you chose Assembler or Structured Macro over a high level language, what is it specifically that you don't like about high level languages?

What things don't you like about the languages you have used?

5 Would you be willing to spend a half hour with me to discuss programming languages?

Would you please return this questionaire to me when you've completed it. I'm on the second floor at Roosevelt, and my telephone extension is 289. Additional copies of this questionnaire can be picked up from Kathleen Pickard in the Basement at Roosevelt.