1975

The Dependence of Operating System Size Upon Allocatable Resources

Atilla Elci

Report Number:
75-172

The Dependence of Operating System Size Upon Allocatable Resources

Atilla Elci
Computer Science Department
Purdue University
West Lafayette, Indiana 47907

December 1975
CSD-TR 172

ABSTRACT

A detailed analysis of 40 operating systems demonstrates that the number of instructions required to control their resources varies over four orders of magnitude, but that 88% of this variance depends only upon the number of unique types of resources controlled. Differences in the size contributions of different resource types were not detectable.

In addition, a software theoretic model is derived, and shown to be in accord with the exponential nature of the data.

I. INTRODUCTION

In the past, the factors which contribute to the size of an operating system have not been well understood. In fact, even the efforts to obtain procedures for estimating program sizes in general have not produced highly accurate methods [1,2,3,4,5,6]. However, some insight into the processes which might be operating is alluded to by Brooks [2] in relation to the programming team size, and by Birkhoff [7] in relation to computational complexity of combinatorial problems.

Since a wide range of reliable data may often contribute to better understanding of any phenomenon, it is the primary purpose of this paper to present the results of a detailed analysis [8] of some 40 different operating systems, ranging from the smallest which could be found to some of the largest. For each of these systems, the count of unique allocatable resource types was obtained (as defined in Section II), together with a count of those instructions which control them. The high correlation observed between the number of instructions and the number of unique resource types is shown in the statistical analysis of Section III.

Borrowing some concepts and software relationships from a developing research area designated as Algorithm Dynamics or Software Physics, Section IV introduces the derivation of an equation relating the size to the number of distinct resources.

II. DEFINITIONS. THE DATA BASE.

A. Definition of resource types.

In general, any facility of the computing system or the operating system required by a job including hardware and software alike might be recognized as a resource. We revise this definition by a restriction which will disqualify some facilities. A facility can become a resource if the operating system has control over it, or there is at least a mutually responsive protocol established between the facility and the operating system. Thus, our definition of a resource type can be stated as follows:

- A unique physical device or the program library whose control is among the operating system's capabilities.

The following list includes all of the resource types encountered in this study:

- Central facilities:
  1. Central processing unit,
  2. Memory,
  3. System timer (real-time clock),
  4. System console,
  5. The program library,
- Storage or input/output devices:
  6. Tape drive,
  7. Disk unit,
  8. Drum unit,

9. Cassette,
10. Card-random-access memory (CRAM),
11. Teletype terminal,
12. Display terminal,
13. Card reader,
14. Paper tape reader,
15. Keyboard,
16. RJE (CR/LP) station,
17. MICR/OCR,
18. Hand-held numeric/function keyboard,
19. Microscope terminal,
20. Scanning/measuring projector,
21. Buttonboard,
22. Precision CRT,
23. Photomultiplier,
24. Mirror assembly,
25. Line printer,
26. Plotter,
27. Paper tape punch,
28. Scanner,
29. Card punch,

Analog/digital facilities:

30. A/D channels,
31. Discrete output lines,
32. Discrete input lines,
33. Analog input lines,
34. Analog output lines,
35. A/D converter,

- Remote processors:

36. CDC 6500,
37. CDC 6400,
38. IBM 360/40,
39. IBM 360/44,
40. IBM 360/22,
41. IBM 7094,
42. IBM 7750,
43. IBM 1401,
44. PDP 11/45,
45. PDP 1,
46. DGC SUPERNOVA,
47. XDS 9300,
48. MODCOMP IV,
49. AD-4,
50. Elbit 100,
51. Digital filter unit,
52. Polly (μpup)

B. Definition of the size of a CFOS.

- An operating system consists of two bodies of programs:

1. Control functions part, (CFOS), is composed of the routines implementing the control of the system resources, and the communication with the environment. These routines can be collected in three divisions: (i) Job management, (ii) Task management, and (iii) Data management parts;

2. The processing programs part, (PPOS), is composed of language translators, service programs, and problem programs.

For the size of a CFOS, we will choose the number of instructions as the unit of measure for no better reason than that programs implement algorithms whose actions are expressed in instructions. Then, the size of a CFOS is the total additive number of machine instructions of the group of routines which implement the CFOS functions, and the PPOS is ignored.

C. The data base.

Forty operating systems are included in our forty point data base. Each operating system is represented by the two variables defined above: the number of its distinct resources, and the size of its CFOS. Each entry also includes additional information as to the identity of the individual resources available. The data base is displayed in Table 1.

Table 1. The data base collected.

NAME OF OS		NUMBER OF RESOURCES	SIZE (K instr.)	LIST OF RESOURCES*
POLLY	(μpup)	5	.016+	1,2,22,23,46
TLMTR	(IBM 7094)	6	.403	1,2,11,13,25,36
PILDT TSDS	(UNIVAC 1108)	7	.474	1,2,3,4,5,13,25
DIGITAL FILTER LAB.	(PDP 11/10)	8	1.034	1,2,4,7,11,14,18,51
SOS	(DGC 800)	9	2.816	1,2,4,5,6,11,14,25,26
SOS	(DGC 800)	9	2.816	1,2,4,5,9,11,14,25,26
DOS	(IBM 360/30)	10	3.219	1,2,3,4,5,6,7,13,25,29
SNOS	(DGC SUPERNOVA)	10	4.801	1,2,3,4,12,15,21,24,38,52
PCP	(IBM 360/30)	10	4.918	1,2,3,4,5,6,7,13,25,29
ABMS	(PDP 11)	10	5.000	1,2,3,11,14,27,31,32,33,34
RT-11	(PDP 11)	10	5.250	1,2,3,4,5,7,11,12,26,30
TUMTR	(IBM 7094)	10	5.943	1,2,3,6,11,13,25,28,36,43
FOURIER OS	(HP 2100S)	10	6.042	1,2,4,5,11,12,14,15,26,35
RC 4000 MS	(RC 4000)	10	6.200	1,2,3,4,6,7,11,14,25,27
M2ØS	(MODCOMP II)	10	8.058	1,2,3,11,13,25,26,36,40,44
M3ØS	(MODCOMP III)	10	8.058	1,2,3,11,13,25,26,36,40,41
OS/MFT	(IBM 370/145)	10	21.312	1,2,3,4,5,6,7,13,25,29
V5	(PDP 15/30)	11	11.881	1,2,4,5,6,7,11,12,13,14,25
ZANEMAR	(NCR 4130)	11	12.000	1,2,4,5,10,11,13,14,25,27,50
OS/1621	(MICRODATA 1621)	11	13.900	1,2,4,5,6,7,11,13,14,25,27
MINIMOP 2	(ICL 1909)	11	14.000	1,2,3,4,5,6,7,11,14,25,27
DOS 15	(PDP 15/30)	11	14.828	1,2,4,5,6,7,11,12,13,14,25
MAXIMOP	(ICL 1909)	11	15.000	1,2,3,4,5,6,7,11,14,25,27
SDC TSS	(IBM AN/FSQ-32)	11	16.384	1,2,5,6,7,8,12,13,29,43,45

Table 1 cont.

NAME OF OS		NUMBER OF RESOURCES	SIZE (K instr.)	LIST OF RESOURCES*
SAS	(NCR 4130)	11	21.000	1,2,4,5,10,11,13,14,25,27,50
CREOPS	(NCR 4130)	11	28.000	1,2,4,5,10,11,13,14,25,27,50
PS/MFT	(IBM 360/40)	11	29.632	1,2,3,4,5,7,11,19,20,39,46
PS/MFT	(IBM 360/44)	11	31.309	1,2,3,4,5,6,7,13,25,29,38
GEORGE 3	(ICL 1904A)	11	40.000	1,2,4,5,6,7,8,11,13,16,25
MAC SYSTEM	(IBM 7094)	12	32.000	1,2,3,5,6,7,8,12,13,25,29,42
CTSS	(IBM 7094)	12	32.700	1,2,3,5,6,7,8,12,13,25,29,42
CTSS	(IBM 7094)	12	32.768	1,2,3,5,6,7,8,11,12,13,25,29
OS/MVT	(IBM 370/155)	12	42.076	1,2,3,4,5,6,7,11,12,13,25,29
OS/MFT	(IBM 370/155)	12	48.094	1,2,3,4,5,6,7,11,12,13,25,29
RSX 15	(PDP 15/30)	12	49.920	1,2,3,4,5,6,7,11,12,13,14,25
EXEC 8	(UNIVAC 1106)	12	60.000	1,2,3,4,5,6,7,11,12,13,25,29
CP/67+CMS	(IBM 360/67)	13	64.363	1,2,3,4,5,6,7,8,11,12,13,25,29
OS/VS1	(IBM 370/145)	13	109.300	1,2,3,4,5,6,7,11,12,13,17,25, 29
OS/MVT	(IBM 360/65)	14	84.494	1,2,3,4,5,6,7,8,11,13,25,29, 47,49
DUAL MACE	(CDC 6500+6400)	15	129.943	1,2,3,4,5,6,7,13,14,25,27,29 37,41,48

*Note: The numbers refer to the list in Section II.A.

III. ANALYSIS

In search of a relationship between the size and the number of resources we performed a regressional analysis on the data. Polynomial regression produced the second degree polynomial shown in Table II. Inspite of its rather high coefficient of correlation (0.94) this polynomial has a serious drawback: it predicts negative sizes. Non polynomial behavior of the data is also indicated by the sign fluctuations in the forward differences. Nonlinear regression on an exponential model produced the results in Table II.

Table II. Results of regression analysis on the size.

Model	$y(x) = b_0 + b_1X + b_2X^2$	$b_0 X \varepsilon^{b_1 X}$
Regression coefficients:		
b_0	130.7178	.03447
b_1	-36.9427	.37489
b_2	2.4687	—
Coefficient of correlation	.938379	.9128

IV. DERIVATION OF THE SIZE FORMULA

To derive a formula to calculate the size of a CFOS given the number of resources it should control, we make use of concepts and formulae from a developing research area designated as Algorithm Dynamics or Software Physics [9-21].

In Software Physics η_2^* represents the number of input/output parameters to an algorithm. In relation to operating systems, we identify η_2^* with the number of distinct resources a CFOS manages.

By the Second Law of Software Physics, the internal quality of a pure algorithm is defined as:

$$IQ = LV = V^* \tag{1}$$

where level, $$L = \frac{2}{\eta_1} \frac{\eta_2}{N_2}$$

volume, $$V = N \log_2 \eta$$

minimum volume, $$V^* = (2 + \eta_2^*) \log_2 (2 + \eta_2^*)$$

length, $$N = N_1 + N_2$$

vocabulary, $$\eta = \eta_1 + \eta_2$$

and

N_1 is the total usage of operators,

N_2 is the total usage of operands,

η_1 is the number of distinct operators,

η_2 is the number of distinct operands.

Substituting $N \log_2 \eta$ for V, and $\frac{2}{\eta_1} \frac{\eta_2}{N_2}$ for L above

$$V^* = \frac{2}{\eta_1} \frac{\eta_2}{N_2} N \log_2 \eta \qquad (2)$$

Bulut's study [18] of the first fourteen algorithms from ACM Algorithms shows that mean η_2/η_1 ratio for machine language versions is .9875. We will assume that $\eta_1 \cong \eta_2$. In machine language, each instruction carries an operator and an operand. Where the operand is indexed there is one more operator - operand pair. Thus, the total count of operator usage, N_1, goes hand in hand with the total count of operand usage, N_2. However, branch instructions add only to N_1 but not to N_2. Similarly, subroutine call parameters placed in-line add to N_2 but not to N_1, thus partially compensating for branch instructions. Consequently, the N_1/N_2 ratio comes quite close to 1.0. Substituting these two assumptions into 2, and solving it for η we obtain

$$\eta = 2^{\frac{V^*}{4}} \qquad (3)$$

It has been empirically validated [21,22,23] that

$$N = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2 \qquad (4)$$

Making use of $\eta_1 \cong \eta_2$ assumption, and substituting 3 into 4 we obtain

$$N = (V^* - 4) \frac{2}{4}^{\frac{V^*}{4}} \qquad (5)$$

Knuth's study [24] of FORTRAN programs shows that:

1. 78% of operands are variables, and
2. 42% of variables are indexed.

It is clear that these statistics are carried over to the machine language versions of the FORTRAN programs studied. Assuming that the statistics above hold for machine language programs, we will employ them to relate N to the number of instructions, P. Specifically,

$$N = 2(1 + .42 * .78)\ P$$

$$\simeq \frac{8}{3}\ P \qquad (6)$$

Combining 5 and 6, we can solve for P to obtain

$$P = \frac{3}{32}\ (V^* - 4)\ 2^{\frac{V^*}{4}} \quad \text{number of instructions} \qquad (7)$$

In the formula 7, V^* can be replaced by $(2 + \eta_2^*)\ \log_2\ (2 + \eta_2^*)$ to calculate the size of a CFOS, P, given the number of distinct resources, η_2^*, it should manage.

V. COMPARISON

In Figure 1 the observed and the theoretical sizes are plotted against the number of resources. (The observed size is the size of a CFOS as it is found in the data base. The theoretical size is the size, P, obtained from the formula 7 above). For ease of comparison the means of the data at each number of resources group are connected. We observe that the curve formed by the means of the data follows the theoretical-size curve closely. To determine how close the agreement is we resort to a regression analysis.

After replacing P in formula 7 above by the size of a CFOS in our data base, we can solve 7 for n_2^* to obtain n_2^*-calculated, the calculated-number of resources. Together with the observed n_2^*, n_2^*-calculated forms a couple for each of the points in our data base which we can correlate. Simple linear regression of n_2^*-calculated versus n_2^*-observed produces .94 as the coefficient of correlation.

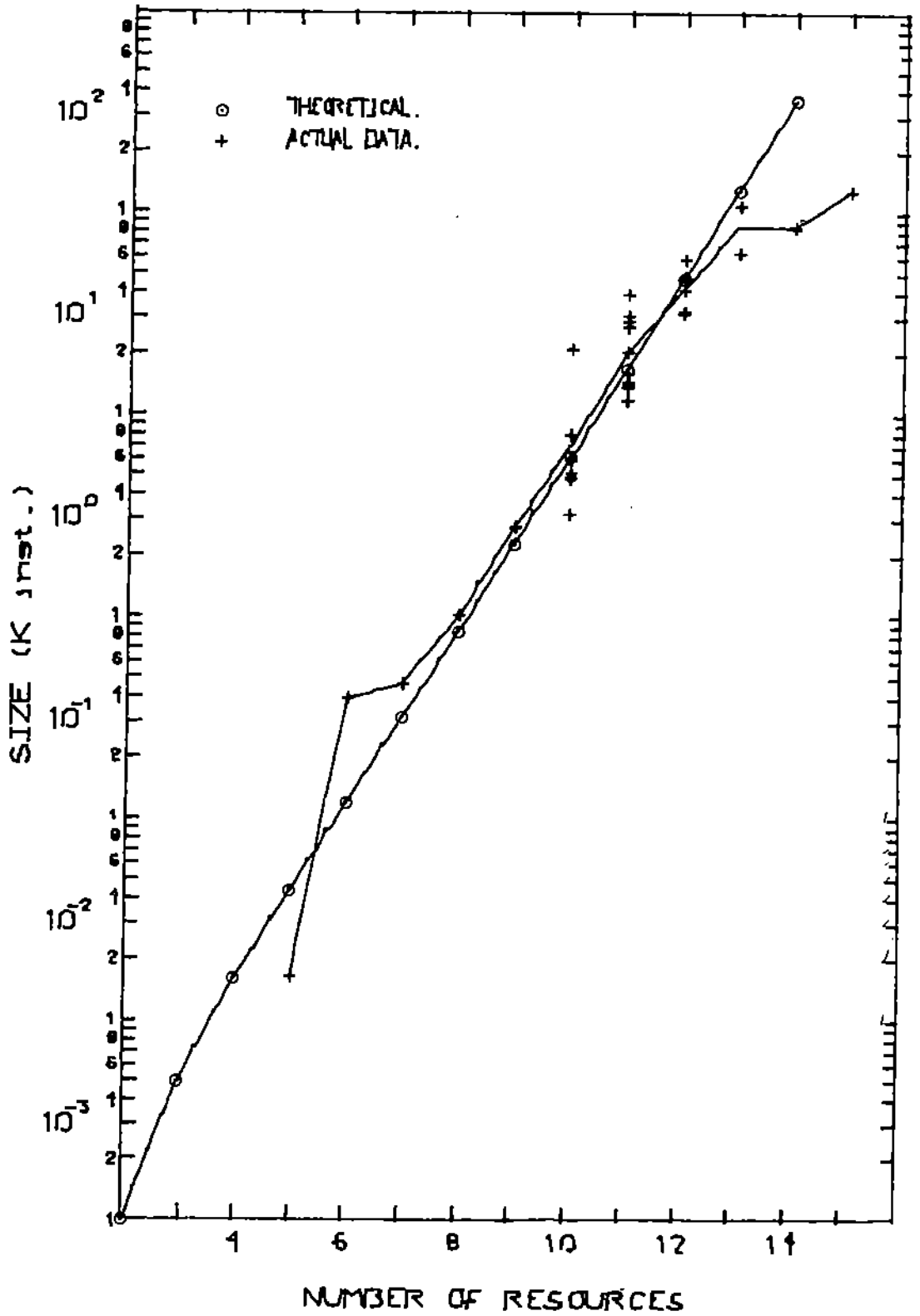

Figure 1. Observed and theoretical sizes versus number of resources.

VI. CONCLUSION AND DISCUSSION

In Section III, statistical analysis of the data indicated non-polynomial characteristics. On the other hand, an exponential model was shown to fit the data comfortably. The predictions of the theoretical-size formula derived in Section IV, which is also exponential, compared favorably with the data yielding a high correlation.

Thus it may be concluded that:

1. The study has confirmed our hypothesis that as more and more resources are incorporated into the configuration of a computing system, the size of its CFOS increases exponentially;

2. The resources are of equal potency in determining the size of a CFOS, and what counts is the presence of a resource in the configuration;

3. The size formula derived is a valid model for the size of a CFOS given the number of resources CFOS manages.

As a result of the study, a model for the size of a CFOS is derived where none existed before. In general terms the model states that:

1. The size of a CFOS is determined primarily by the number of distinct resources CFOS manages;

2. The relationship between the size and the number of resources is of exponential character;

3. In determining the size, the identity of the individual resources is immaterial, and what matters is that CFOS pays attention to the individual resources simultaneously and in combination with all of the others.

The model provides a fairly basic understanding of the processes involved in CFOS size estimation as well as a quantitative method. However, the size formula is presented as an integral part of the model. It should be employed in the context of the model and not as a substitute for it. We can identify two pitfalls regarding the usage of the size formula. First, since the formula is dependent on the number of resources, a proper identification of distinct resources is crucially important. Failure to do so may lead to exaggerated size estimates. Secondly, the size estimate given by the formula should not be taken as precise and without its margin of variation. If we let P_n be the evaluation of the size formula for n resources, the margin of variation associated with P_n can be stated as from P_{n-1} to P_{n+1}. The importance of this context can be vividly visualized if one realizes that the margin around P_n is at least three times P_n. (Note that the size formula indicates $P_i > 2.68 P_{i-1}$). While other CFOS design methods may yet be invented to lower the size from what the size formula predicts, in our data base, the encountered reductions in size were not large enough to cut the size to less than P_{n-2} for any n - resourced CFOS. Similarly, the design strategy used in Dijkstra's THE - Multiprogramming System [25], insofar as it is fairly represented by Brinch Hansen's RC 4000 Multiprogramming System [26,27], was found incapable of achieving any important reduction, since in the latter case the observed length was 6200 instructions against 6319 predicted by equation 7.

ACKNOWLEDGEMENTS

I thank Professor M. H. Halstead for wisdom, guidance, and enduring patience. I also thank Professors S. D. Conte and H. D. Schwetman for their constructive suggestions. My appreciations are due to the many individuals who helped us, both directly and through correspondence in obtaining values and listings for our data base.

LIST OF REFERENCES

1. HIRSCH, R. E., "Programming performance: Monitoring, maximization, and prediction", Proc 10th SIG CPR C 1972, pp 36-46.

2. BROOKS, F. P., Jr, The Mythical Man-Month, Essays on Software Engineering, Addison-Wesley, 1975.

3. HENNEY, A., "Techniques for estimating programming effort and assessing programmer performance", Data Systems (Sept 1969), pp 34-36.

4. SHELL, R. L., "Work measurement for computer programming operations", Industrial Engineering, (October 1972), pp 32-36.

5. ARON, J. D., "Estimating resources for large programming systems", Software Engineering Techniques, 1969, (Buxton and Randal, ed.'s), pp 68-79.

6. PIETRASANTA, A. M., "Resource analysis of computer program system development", in On The Management of Computer Programming, (Weinwvrm, G. F., ed.), Averbach 1970.

7. BIRKHOFF, G., "Mathematics and Computer Science", American Scientist, 63, (January-February 1975), pp 83-91.

8. ELCI, A., Factors Affecting The Program Size of Control Functions of Operating Systems, Ph.D. Thesis, Purdue University, December 1975.

9. HALSTEAD, M. H. A Thermodynamics of Algorithms?, CSD TR 66, Purdue University, Computer Science Department, February 1972.

10. _____, "Natural Laws controlling algorithm structures?", ACM-SIGPLAN Notices, 7,2 (February 1972), pp 19-26.

11. _____, A Theoretical Relationship between Mental Work and Machine Language Programming, CSD TR 67, Purdue University, Computer Science Department, February 1972.

12. _____, "An experimental determination of the 'purity' of a trivial algorithm", ACM-SIGME:Performance Evaluation Review, 2,1 (March 1973), pp 10-15.

13. _____, "Language level, a missing concept in Information Theory", Ibid, pp 7-9.

14. BAYER, R., A Theoretical Study of Halstead's Software Phenomenon, CSD TR 69, Purdue University, Computer Science Department, May 1972.

15. HALSTEAD, M. H., and R. BAYER, "Algorithm Dynamics", Proc. ACM National Conference 1973, pp 126-135.

16. ZWEBEN, S. H., Software Physics: Resolution of an Ambiguity in The Counting Procedure, CSD TR 93, Purdue University, Computer Science Department, April 1973.

17. HALSTEAD, M. H., and P. M. ZISLIS, Experimental Verification of Two Theorems of Software Physics, CSD TR 97, Purdue University, Computer Science Department, June 1973.

18. BULUT, N., Invariant Properties of Algorithms, Ph.D. Thesis, Purdue University, August 1973.

19. ZWEBEN, S. H., The Internal Structure of Algorithms, Ph.D. Thesis, Purdue University, May 1974.

20. HALSTEAD, M. H., and N. BULUT, "Impurities found in algorithm implementations", ACM-SIGPLAN Notices, 9,3 (March 1974), pp 9-12.

21. BULUT, N., M. H. HALSTEAD, and R. BAYER, "Experimental validation of a structural property of FORTRAN algorithms", Proc ACM National Conference 1974, pp 207-211.

22. ELSHOFF, J., "Measuring commercial PL/I programs using Halstead's criteria", GM Research Report, and also in ACM SIGPLAN Notices, February 1976.

23. BOHRER, R., "Halstead's criteria and statistical algorithms", Proc. 8th Computer Science/Statistics Interface Symposium, Los Angeles, February 1975.

24. KNUTH, D. E., "An empirical study of FORTRAN programs", Software-Practice and Engineering, 1,2, 1971.

25. DIJKSTRA, E. W., "The structure of THE - multiprogramming system", CACM 11, 5 (May 1968), pp 341-346.

26. BRINCH HANSEN, Per, "The nucleus of a multiprogramming system", CACM 13, 4 (April 1970), pp 238-241, 250.

27. BRINCH HANSEN, Per, Operating System Principles, Prentice-Hall, 1973. (Chapter 8).