UNDERSTANDING COMPUTER SYSTEMS

HAROLD W. LAWSON, JR.

COMPUTER SCIENCE PRESS

UNDERSTANDING COMPUTER SYSTEMS

HAROLD W. LAWSON, JR.
Linköping University, Sweden

COMPUTER SCIENCE PRESS

Printed in the United States of America.

Computer Science Press, Inc.
11 Taft Court.
Rockville, Maryland 20850

1 2 3 4 5 6 Printing Year 86 85 84 83 82

Library of Congress Cataloging in Publication Data

Lawson, Harold W.
Understanding computer systems.

1. Electronic digital computers. I. Title.
QA76.5.L373 1982 001.64'2 81-17295
ISBN 0-914894-31-5 AACR2

Cover design by Ruth Ramminger

PREFACE

The impact of computer systems technology upon the individual and society has steadily increased since the introduction of modern digital computers after World War II. Computers have become more powerful and less expensive and this has led to a widespread governmental, institutional, industrial and business utilization of the technology. In particular, the development of "microcomputers" in the 1970's has resulted in the possibility for small organizations and, in fact, most individuals to own and operate their own computer.

This growing impact most certainly is resulting in a "need to know" more about the technology as well as its impacts upon the citizenry of the world. In Norway, the labor unions have obtained agreements with employer organizations concerning computer based systems. The agreements contain paragraphs that express the **right** of employees to "gain insight into and understand fundamental features of the systems **they themselves use or are affected by** and to understand the importance of the use of such systems both to the company and the employees in their work situation". Further, the agreement states that systems must be explained in a "language easily understood by persons lacking special knowledge of the area concerned". A growing awareness and, in fact, revolutionary attitudes similar to those related to atomic energy can occur in relationship to the widespread utilization of computer based systems, particularly when job replacement by automation occurs. Education based upon an easy to understand approach to computer systems as well as timely, responsible political and socio-economic decisions are essential to achieving a peaceful transition to the automation of organizational activities.

Unfortunately, while major advances in computer systems have been accomplished, corresponding advances in providing for an easy understanding of the technological aspects of computer systems have not been made. Today's systems are, to a large extent, as difficult to understand and utilize as the earliest computers. Computer manufacturer supplied literature, in the form of manuals, are often difficult to understand, particularly for the beginner, and an overview of the computer system as a whole is quite difficult to ascertain. Most textbooks on the subject begin by introducing the details of how the central parts of a computer system work and the reader "student" studies the trees and their leaves instead of first obtaining an overview of the forest.

This book is directed, firstly, to beginners who know nothing or very little about computer systems. The novel approach used here will give the reader "student" an overview of computer based systems including important concepts and terminology. This overview provides a solid platform upon which to study the details of computer technology for those who plan to go further and specialize in the field as well as providing the layman with a well rounded view of the computer technology milieu. Further, the book is directed to those who already have computer technology knowledge as a guide to solidifying the many aspects of the technology as well as providing assistance in educating others.

The book may, for example, be utilized in an introductory course in computer science or computer technology to give a brief but comprehensive overview of what is to come. The book may also be effectively utilized in a computer appreciation course or study-circle since all participants can participate in discussing the subject material directly from the beginning. Teaching assistance materials for utilizing this book in the classroom or in study-circles are available.

I have wanted to write this book for many years and finally, I have found time to collect my thoughts and present them. Any effort of this type is not the work of one individual and I wish to express my thanks to several "interested parties." First, to my many colleagues in many corners of the world from whom I have learned and with whom I have had the opportunity to discuss the novel approach followed in this book. A special thanks to Captain Dr. Grace M. Hopper for providing the starting point for my own base of knowledge in the field of computing. Secondly, to the students and staff at Linkopings University who have read, commented and corrected. Particularly, I would like to thank Rolf Flisberg, Bjorn Gudmundsson, Bengt Johnsson and Erik Tengvald for their very useful comments. Thirdly, to Marianne Anse-Lundberg, Tomas Hedblom, Eleanor Johansson and Lena Hjaellsten who helped with drawing and typing. Fourthly, to Peter Jensen and Richard Eckhouse for providing valuable comments on the first edition of the book. Finally, to Ann and Nadia who provide my inspiration and to whom I dedicate this work.

Happy reading,

Harold W. Lawson, Jr.

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

In attempting to understand the function of computer systems and their utilization, one is faced with many learning difficulties due to the seemingly complicated structure and many intricately interrelated aspects of computer systems. However, in this book we introduce a means of describing computer systems that relates to well-known "real life analogies." By utilizing this associative approach, readers will be able to extend their knowledge of what they already know to include many important computer system concepts and terminology.

In the brief history of digital computer systems (*) there has arisen a need for many types of specialists in the computer field. Unfortunately, as the technology developed, the specialists worked, far too often, in isolation from each other and thus many specialized forms of computer technology related skills developed. This is vividly reflected in the special types of technical "languages" utilized by these specialists in carrying out their work. The interaction of the many technical disciplines required in designing, constructing and utilizing computer systems has led to many complexities. The complexities in computer system structures, the language aspect and the problem of understanding are quite nicely summarized in the following quotation:

> "THE UNIVERSE AND ITS REFLECTION IN THE IDEAS OF MAN HAVE WONDERFULLY COMPLEX STRUCTURES. OUR ABILITY TO COMPREHEND THIS COMPLEXITY AND PERCEIVE AN UNDERLYING SIMPLICITY IS INTIMATELY BOUND WITH OUR ABILITY TO SYMBOLIZE AND COMMUNICATE OUR EXPERIENCE. THE SCIENTIST HAS BEEN FREE TO EXTEND AND INVENT NEW LANGUAGES WHENEVER OLD FORMS BECAME UNWIELDY OR INADEQUATE TO EXPRESS HIS IDEAS. HIS READERS, HOWEVER, HAVE FACED THE DOUBLE TASK OF LEARNING HIS NEW LANGUAGE AND THE STRUCTURES HE DESCRIBED. THERE HAS, THEREFORE, ARISEN A NATURAL CONTROL: A **WORK OF ELABORATE LINGUISTIC INVENTIVENESS AND MEAGER RESULTS WILL NOT BE WIDELY READ.**"
>
> WILLIAM M. McKEEMAN

(*) Note that the first fully electronic digital computer system called ENIAC was designed by J. Presper Eckert and John Mauchly and constructed at the Moore School of Electrical Engineering of the University of Pennsylvania in 1945.

The complexities reflected in the variety of languages of the specialists in the computer field has been a deterent to widespread understanding of computer systems and their applications. Further, several languages in the computer field can be criticized for their linguistic inventiveness and meager results and thus have created barriers to understanding.

The old adage that a picture is worth a thousand words has been applied in describing complex structures of many forms, for example, building architectures, urban development plans, sewage systems, electrical circuit wiring to name a few. This is not, of course, any different in the computer field where specialists develop graphical representations to convey meaning. Pictures are a form of language which have their origins in the earliest history, for example, Egyptian hieroglyphics. Through pictures, understanding can be obtained more rapidly than with natural language text or with specialized technical languages.

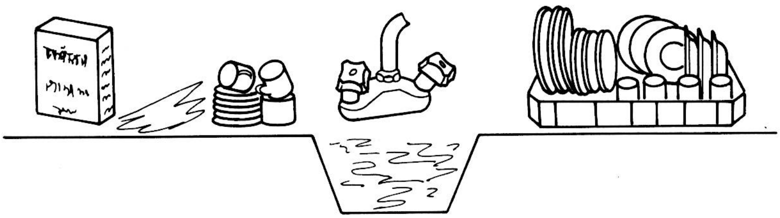
In this book, we shall exploit graphical (pictorial) representations as a means of introducing the structure of computer systems. Further, as mentioned earlier, we will make many real life **analogies** to the type of work carried out at the various technical levels of computer systems; namely, we will look at various real life processes and systems and their counterparts in computer systems. This means that although we introduce many complicated concepts and terminologies, the reader will be introduced to computer system ideas in terms that the reader already understands (pictures and analogies to well known real life processes).

In the field of computing, there are many words used to name concepts and technical components. Further, the names frequently have **synonyms** which are equivalent terms. These synonyms are frequently due to different groups that have developed the same general concepts. By and large, most of the computer related technical terminology has been derived from the English language. However, as with other minor differences between English dialects, there are American based terms and British based terms for computer concepts and components. In any event, when a concept or component is introduced for which one or more synonyms exist, this existence will be clearly indicated in the text.

Chapter 2

UNDERSTANDING PROCESSES AND SYSTEMS

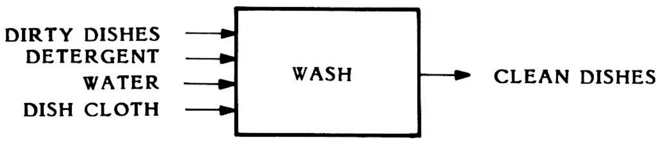
We begin this education process with a pictorial representation of a well-known real life **process (task)** faced by many of us.



A PROCESS

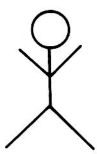
Note that the concept of **process** and **task** are synonymous and thus in all further references to process related concepts, the word task may also be utilized.

Let us now consider this real life process of washing the dishes in the form of an abstraction which shows the major elements of this process as follows:



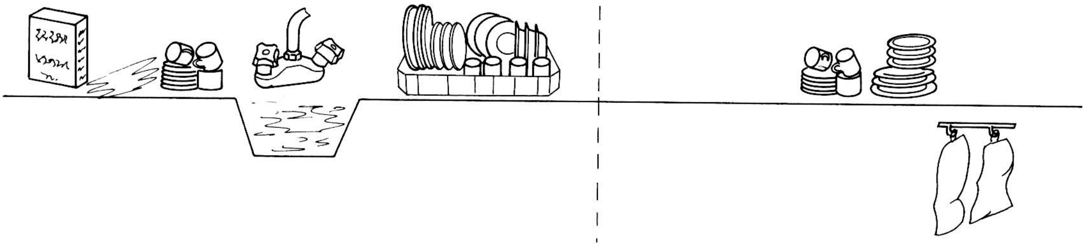
A PROCESS ABSTRACTION

Dirty dishes, detergent, water and dish cloth are **process inputs** and clean dishes are a **process output.** The process which we have shown here can only be carried out **"executed"** when we apply a **processor.**



A PROCESSOR

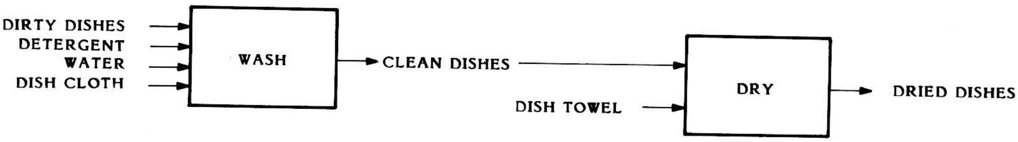
Let us complement this single process by introducing a second process, thus creating a **system** of **cooperating processes.**



A SECOND PROCESS

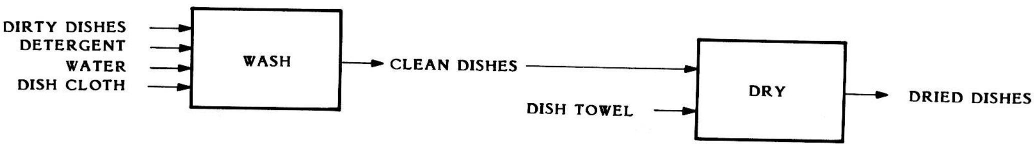
Note: Implicit in this description is the definition of a **system** as a collection of interrelated processes.

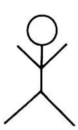
We can continue the general abstraction of this system of cooperating processes by introducing an abstraction for the second process, namely the drying process, in the following manner:



AN ABSTRACTION OF A SYSTEM OF COOPERATING PROCESSES

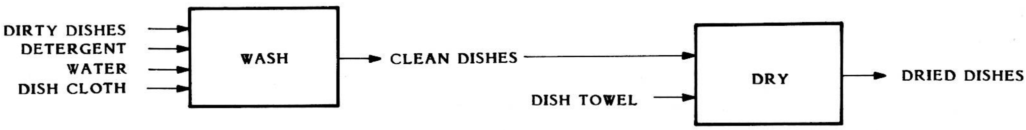
If the processes are to be carried out by a single processor, **uni-processor,** then this single processor must be assigned to both the WASH and the DRY process as indicated in the following picture.





A UNI-PROCESSOR

Note that if the dishrack becomes **full** during WASH execution then the single uni-processor must be alternated between the execution of the processes WASH and DRY. Alternatively, we could assign a processor to each process, that is, the processes are executed **concurrently** (at the same time) by **multi-processors.**

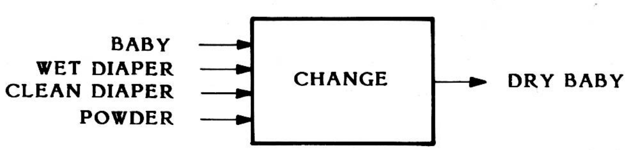


MULTI-PROCESSORS

During execution of a process by a processor, let us say the WASH process, the processor could be **interrupted** by a **higher priority process** such as the following:

A HIGH PRIORITY PROCESS

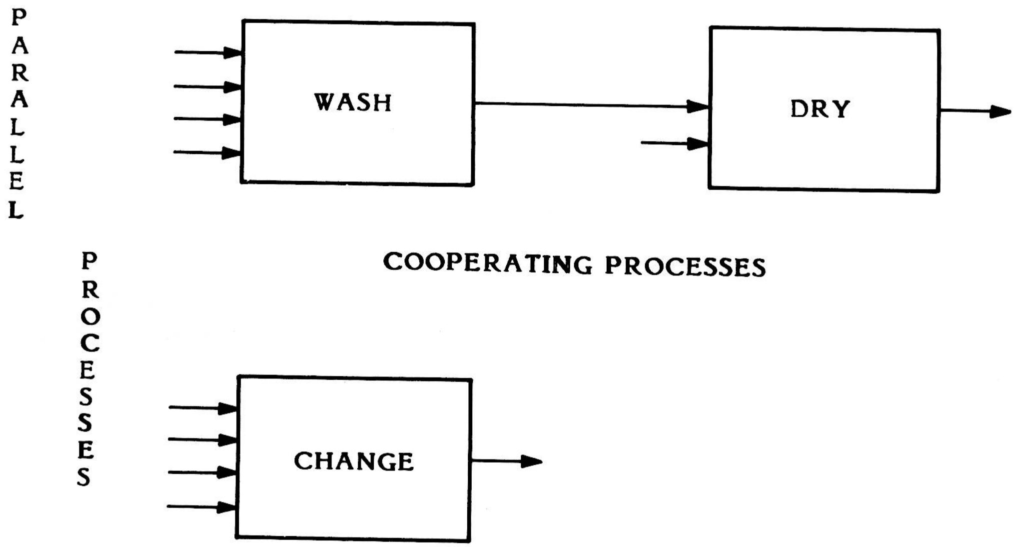
Consequently, a new process is **"created"** which we can represent abstractly as follows:



AN ABSTRACTION OF THE NEW PROCESS

In the case of a single uni-processor, the processor temporarily **"suspends"** the process it is currently executing and **"initiates"** execution of the CHANGE process. After **"termination"** of the CHANGE process, the processor returns to **"resumes"** execution of the process that was suspended.

Alternatively, in the case of multi-processors, the processor that acknowledges the interrupt (let us say the processor serving the WASH process) could **"assign"** another processor (the processor assigned to DRY or another available processor) to execute the CHANGE process in **"parallel"** with the ongoing WASH and/or DRY processes as follows:



Summary

In summary, thus far, if we have a single processor **uni-processor,** the processor can only be assigned to a single process execution at any given moment. The processor can service one of the **cooperating processes** WASH or DRY or if a higher priority process **interrupt** appears (baby cries due to wet diapers), the processor's attention can be turned to executing a **parallel process,** namely the CHANGE process and when that process is completed, it can resume with the WASH or DRY process execution that was temporarily suspended.

If **multi-processors** (let us say two processors) are available, they could, for example, be assigned to the WASH and DRY **cooperating processes** and one of them could be assigned at the time of **interrupt** to the **parallel process** CHANGE while the other processor can continue serving the WASH and/or DRY process.

Word List

We have now considered some rather non-trivial concepts of well-known processes and systems which have their correlaries in computer systems. Readers should verify their understanding of these concepts by once more relating the following terminology to the examples given and to the problems listed below. Please note once again that process and task, which is indicated in parenthesis (), are synonymous terms.

- process (task)
- process inputs
- process outputs
- process execution
- processor
- system
- cooperating processes
- uni-processor
- concurrent processes
- multi-processors
- interrupt
- process priority
- process creation
- process suspension
- process initiation
- process termination
- process resumption
- processor assignment
- parallel processes

Problems

1. Building upon the previous example:

a). Extend the picture of the WASH and DRY processes to include a process for TABLE CLEARING and a DISH STORING process.

b). Extend the abstract description of this system of processes.

c). Explain how the dishes will be processed through these four processes if only a uni-processor is available to execute processes.

d). Explain how the dishes can be processed if we have multi-processors, namely, 2, 3 or 4 processors available to assign to execution of the processes.

e). What happens if the telephone rings while the dishes are being processed, in the case of a uni-processor and in the case of multi-processors?

2. You have been fortunate to receive, as a present, an automatic dishwashing machine.

a). Create a new picture of the system of processes by replacing the WASH and DRY processes by a new single process WASH/DRY.

b). Create a new abstract description reflecting this substitution.

c). Which processor will be applied in executing the WASH/DRY process?

d). How will the processor initiate the execution of the WASH/DRY process?

WARNING

TO THOSE WHO ONLY HAVE A HAMMER, THE WHOLE WORLD LOOKS LIKE A NAIL.

TO THOSE WHO ONLY KNOW COMPUTERS, THE WHOLE WORLD LOOKS LIKE AN INFORMATION PROCESSING MACHINE.

JOSEPH WEIZENBAUM

We have utilized well known real life processes to illustrate some important computer related concepts. It would be a disastrous mistake to assume that all real life processes and systems can be computerized. Our real life natural processes and many processes developed by people and related to individuals and their environment are far too sophisticated and complex to be accurately computerized in every real detail.

Unfortunately, as Weizenbaum suggests, many computing professionals, due to their perspective, narrow their view of the world and try to view everything in terms of equivalents of "computerized" processes and systems. This viewpoint is **not** to be taken by the readers of this book.

With these warnings in mind, let us proceed with obtaining an understanding of computer systems via pictures and real life analogies.

Chapter 3
PROCESS AND SYSTEM "DATA" FLOW

A primary goal of **information** processing systems is to process **data.** But what is **information** and what is **data**? For a definition of data, let us consider the following quotation:

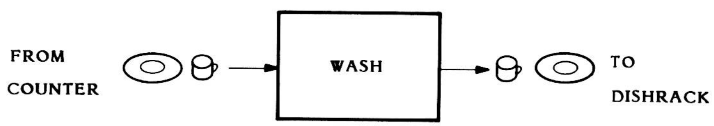
> "DATA: A REPRESENTATION OF FACTS OR IDEAS IN A FORMALIZED MANNER CAPABLE OF BEING COMMUNICATED OR MANIPULATED BY SOME PROCESS."
>
> PETER NAUR

Data as alluded to by Naur is simply a representation of something; whereas, **information** is an interpretation given to data in some well defined context. The numbers (6, 3, 1, 12, 9) are data representations but they only become information when we know what they represent, for example, apples, age, meters, etc.. This point is often confusing and we see references to both **information processing systems** and **data processing systems.** It is a matter of taste as to which term best describes the use of computer systems in processing data/information. We shall utilize the term **information processing system** in this book.

Our real life analogy of the previous chapter is a **dish processing system** in which we deal with real objects instead of representations. However, in continuing our analogy, we shall treat these objects as being counterparts to data in an information processing system.

In formalizing the notion of objects (data) for our dish processing system, let us concentrate only upon the dishes (ignoring detergent, water, etc.) to be processed through the entire system. In order to describe and enumerate all of the possible "types" of dishes that can be processed, we build an **alphabet** of dishes. This alphabet along with an illustrative picture of the objects (data) entering and leaving the WASH process is as follows:

ALPHABET OF DISHES {CUP, SAUCER, SALAD DISH, MEAT DISH}



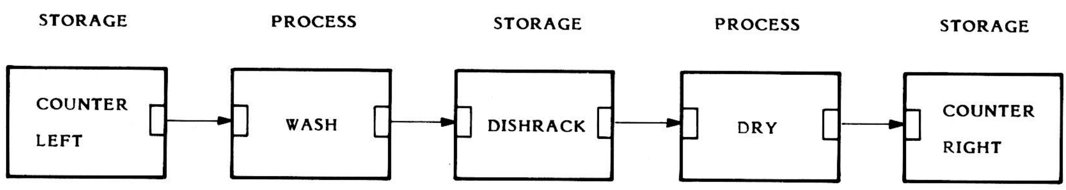
ENUMERATION OF PROCESS RELATED OBJECTS (DATA)

Alphabets in the case of this formalism restrict the type of objects (data) that can be processed by a process. This is completely analogous to the notion of an alphabet in a natural language, let us say English with its 26 letters (A to Z) which then restricts our "word construction process" to using only these 26 alphabet members.

When the objects (data) are processed by a process, they frequently go through a rather standard 3 step cycle which we illustrate in general terms and concretely in terms of the WASH process.

PROCESS STEPS	WASH PROCESS STEPS
input object (data)	take object from counter
processing of object (data)	wash object
output object (data)	place object in dishrack

Now that we have considered the basic concepts of the alphabet of process objects (data) and their flow through a process, let us ask the question: where are the process objects (data) taken from (input) and where are they placed (output)? In the previous picture and process step description, we note that they are taken from a counter and they are placed in the dishrack. Thus the counter and the dishrack are **storages** which hold objects (data) to be processed and during process execution, objects (data) are taken from a storage for processing and placed in a storage after processing. In the following picture, we show an abstraction of the storages, processes and object (data) flow in our system for dish processing.



STORAGES, PROCESSES AND OBJECT (DATA) FLOW

In this new version of our model of the dish processing system where only the flow of objects (data) of the type dishes is illustrated, we assume that the same alphabet of dishes is "known" to and used by all storages and processes in the system. That is, the processes are restricted to processing the same type of dishes.