

# SYSTEM AND SYSTEM PART OF USER PROCESS INITIALIZATION

SEPT 28 1970

## 1) INTRODUCTION

THIS IS A 2 PHASE PACKAGE, THE FIRST PHASE INITIALIZES THE SYSTEM AND CONSTRUCTS GLOBAL OBJECTS. THE RESULTS OF THE FIRST PHASE INCLUDES A DIRECTORY FOR THESE GLOBAL OBJECTS. THE SECOND PHASE INITIALIZES A SINGLE USER PROCESS, IT IS REPEATED FOR EACH INDIVIDUAL USER PROCESS, ALSO CONTROL PROCESSES. *AND*

EACH PHASE MAKES REFERENCE TO A DIRECTORY: THIS DIRECTORY COMES IN TWO PARTS, A GLOBAL DIRECTORY AND A LOCAL DIRECTORY. THE GLOBAL PART IS CONSTRUCTED DURING THE SYSTEM INITIALIZATION PHASE AND REMAINS UNCHANGED THEREAFTER. THE LOCAL PART IS CONSTRUCTED FRESH FOR EACH PROCESS BEING CONSTRUCTED, ( A LOCAL PART ALSO EXISTS DURING SYSTEM INITIALIZATION. ) THE LOCAL PART GROWS DURING PROCESS CONSTRUCTION, ( ALSO DURING SYSTEM INITIALIZATION. ) THE LOCAL PART IS DESTROYED WHEN THE PROCESS IS DESTROYED, OR AT LEAST AFTER THE PROCESS IS INITIALIZED. *ence* EACH PART OF THE DIRECTORY CAN BE THOUGHT OF AS A SEQ OF PAIRS, EACH PAIR IS 1 60 BIT WORD ( THE NAME ) AND 1 CAPABILITY ( THE OBJECT. ) WHENEVER A SPECIFIC NAME IS GIVEN IN THIS DOCUMENT, IT WILL BE REPRESENTED IN DISPLAY CODE, LEFT JUSTIFIED, ZERO FILLED. THROUGHOUT THIS PACKAGE THERE IS A COMMON MEANS OF OBTAINING AN ENTRY FROM THE DIRECTORY, CALLED FETCH, AND COMMON MEANS OF PLACING ENTRIES IN THE DIRECTORY PARTS. THESE ARE DESCRIBED BELOW.

EACH PHASE ALSO MAKES USE OF A SINGLE FORM OF SYSTEM SUBPROCESS DESCRIPTOR. THIS IS A SEQUENCE OF WORDS OCCURING AT THE BEGINNING OF A FILE AND IS DESCRIBED BELOW.



## I SYSTEM INITIALIZATION PHASE

SYSTEM INITIALIZATION IS CONTROLLED BY IPROC AND PROCEEDS AS FOLLOWS:

1) THE GLOBAL DIRECTORY IS INITIALIZED AND SOME INITIAL OBJECTS ARE PLACED IN IT. THESE OBJECTS INCLUDE:

- A) ALL ITEMS OCCURRING IN OPNAMES,XTEXT. EACH IS GIVEN THE NAME I.#NAME# WHERE #NAME# IS THE SAME AS APPEARS IN C.#NAME# IN OPNAMES,XTEXT.
- B) THE MASTER C-LIST ( IPROC#S C-LIST ) AS I.MASTER .  
( ALSO OCCURS AS #I.SELF# )
- C) AN OPERATION FOR CREATING A USER OR CONTROL PROCESS,  
AS #B.NEWPROC# .
- D) THE CLASS CODE TO BE USED FOR THE ROOT OF CONSTRUCTED  
USER OR CONTROL PROCESSES, AS #B.USERROOT#

2) THE FIRST FILE ( FOLLOWING ECS SYSTEM ) IS THEN READ FROM THE TAPE. ( WE CALL IT THE NAME FILE IN WHAT FOLLOWS. )

3) THE NAME FILE IS NOW SCANNED STARTING WITH IT#S FIRST WORD.

- A) IF THE WORD IS NON ZERO, A FILE IS READ IN OFF THE TAPE AND PLACED IN THE GLOBAL DIRECTORY WITH NAME THE NON ZERO WORD SCANNED.
- B) IF THE WORD IS ZERO. THIS STEP IS COMPLETED.

4) THE LOCAL DIRECTORY ( THE ONE TO BE USED DURING INITIALIZATION ONLY, ) IS NOW INITIALIZED TO INCLUDE:

- A) ONE CLASS CODE AND ONE ZERO PARAMETER OPERATION USING THAT CLASS CODE, CALLED RESPECTIVELY #CLASS# AND #CALL# .
- B) THE CLASS CODE FOR THE ROOT SUBPROCESS, CALLED #ROOT# .

5) THE NAME FILE ~~IS~~ IS NOW SCANNED STARTING WITH THE 1ST WORD AFTER THE 1ST ZERO WORD.

- A) IF THE WORD IS NON ZERO, IT IS TAKEN AS THE NAME OF A FILE. THIS NAMED FILE IS FETCHED FROM THE DIRECTORY AND USED AS A SUBPROCESS DESCRIPTOR TO BUILD A SUBPROCESS. ( SEE BELOW. ) THE CONSTRUCTED SUBPROCESS IS THEN CALLED, ( USING IT#S INITIALIZATION CALL. ) WITH RETURN PARAMETER AUTHORIZATION, BOTH DATA AND CAPABILITY. UPON RETURN THE DATA PARAMETERS ARE SCANNED FROM THE BEGINNING. IF NON ZERO THE WORD IS TAKEN AS A NAME AND THE PAIR CONSISTING OF THAT NAME AND THE RETURNED CAPABILITY WITH THE SAME RELATIVE INDEX ARE PLACED IN THE GLOBAL DIRECTORY. IF ZERO THE RETURN PARAMETER SCAN IS STOPPED, THE SUBPROCESS IS DELETED, THE SCRATCH FILE AND LOCAL C-LIST FOR THE SUBPROCESS ARE DESTROYED. ( BUT THE ENTRIES PLACED IN THE LOCAL DIRECTORY FOR THESE LAST TWO OBJECTS ( SEE BELOW ) ARE NOT DELETED. BEWARE! )

B) IF THE WORD IS ZERO, THIS STEP IS COMPLETED.

6) SYSTEM INITIALIZATION IS NOW COMPLETE AND IPROC REWINDS AND UNLOADS THE TAPE, AND THEN PROBABLY DIES. ( THIS LAST NOT YET DECIDED. )



17 CONSTRUCTION OF A USER OR CONTROL PROCESS.  
(- USED TO BE CALLED PROCESS INITIALIZATION PHASE )

1) ANY SYSTEM SUBPROCESS DURING INITIALIZATION OR WHILE RUNNING IN ANY PROCESS CONTAINING A BUILDER MAY CAUSE CREATION OF ANOTHER PROCESS CONTAINING A BUILDER.

2) TO CREATE THIS PROCESS, AN OPERATION UNDER THE NAME #B.NEWPROC# EXISTS IN THE GLOBAL DIRECTORY. IT HAS 3 PARAMETERS.

A) DATUM A NAME THAT WILL BE FETCHED AND TREATED AS A FILE OF NAMES OF SUBPROCESS DESCRIPTORS ( NAME FILE ) DURING INITIALIZATION OF THE PROCESS.

B) BLOCK DATA A SEQ OF NAMES TERMINATED BY A ZERO WORD. TO BE USED FOR INITIALIZATION OF THE NEW PROCESS'S LOCAL DIRECTORY. FOR EXAMPLE, TTY OBJECTS WOULD BE PASSED THIS WAY.

C) BLOCK CAPABILITY A SEQUENCE OF CAPABILITIES EQUAL IN LENGTH TO THE NUMBER OF NAMES PLACED IN B). THESE CAPABILITIES WILL BE PLACED IN THE NEW PROCESS'S LOCAL DIRECTORY.

3) #B.NEWPROC# CALLS THE BUILDER SUBPROCESS IN THE LOCAL PROCESS. THIS BUILDER THEN CREATES AND INITIALIZES A LOCAL DIRECTORY FOR THE NEW PROCESS. IT THEN CREATES THE SCRATCH FILES AND CLISTS NEEDED BY THE NEW PROCESS ROOT AND BUILDER. IT PLACES THE NAME OBTAINED AT 2) A) INTO THE NEW BUILDER SCRATCH FILE. THE NEW PROCESS IS THEN CREATED AND ITS ROOT IMMEDIATELY CREATES A BUILDER. WHAT FOLLOWS TAKES PLACE IN THE BUILDER.

4) THE CLASS CODE FOR THE ROOT OF THE NEW PROCESS IS PLACED IN THE NEW LOCAL DIRECTORY UNDER THE NAME #ROOT#.

5) THE NAME PASSED IN THE NEW BUILDERS SCRATCH FILE IS ASSUMED TO BE THE NAME OF A FILE IN THE DIRECTORY, THAT FILE IS FETCHED. ( WE CALL THAT FILE THE NAME FILE IN WHAT FOLLOWS )

6) THE NAME FILE IS SCANNED FROM THE BEGINNING.

A) IF THE WORD IS NON ZERO, IT IS TAKEN AS THE NAME OF A FILE. THIS NAMED FILE IS FETCHED FROM THE DIRECTORY AND IS USED AS A SUBPROCESS DESCRIPTOR TO BUILD A SUBPROCESS ( SEE BELOW. ) THE CONSTRUCTED SUBPROCESS IS THEN CALLED USING ITS INITIALIZATION CALL. NO RETURN PARAMETER AUTHORIZATION. THE SUBPROCESS IS NOT DESTROYED AFTER THE CALL.

B) IF THE WORD IS ZERO, THIS STEP IS COMPLETED.

7) BUILDER NOW DESTROYS ALL CONSTRUCTED SUBPROCESSES, THEIR SCRATCH FILES AND LOCAL C-LISTS, AND RETURNS TO ROOT.

[ THIS LOGIC MAY CHANGE LATER, AS IT IMPLIES THAT THE PROCESS  
RUNS WITH THE BUILDER IN EXISTENCE. ]

[ NOTE THAT THE INITIALIZATION CALL ON THE LAST SUBPROCESS IS  
CONSIDERED TO START THE PROCESS, AND RETURN FROM THAT LAST  
SUBPROCESS DESTROYS THE PROCESS. ]

8) THE ROOT WILL NOW COMMIT SUICIDE.



## I) DIRECTORY ACTIONS

### A) PLACE AN ENTRY IN GLOBAL DIRECTORY:

ENTRY IS PLACED AT END OF GLOBAL DIRECTORY, IRRESPECTIVE OF WHETHER DUPLICATE NAMED ENTRY EXISTS.

### B) PLACE AN ENTRY IN LOCAL DIRECTORY.

ENTRY IS PLACED AT END OF LOCAL DIRECTORY, IRRESPECTIVE OF WHETHER DUPLICATE NAMED ENTRY EXISTS.

### C) FETCH AN ENTRY.

THE 2 DIRECTORY SEQUENCES ARE CONCATENATED TOGETHER, THE GLOBAL DIRECTORY FOLLOWED BY THE LOCAL DIRECTORY, TO FORM A SINGLE SEQUENCE OF PAIRS. THIS SINGLE SEQUENCE IS THEN SCANNED BACKWARDS FROM THE END UNTIL EITHER AN OCCURRENCE OF THE DESIRED NAME IS FOUND, OR THERE IS DETERMINED TO BE NO SUCH OCCURRENCE. IN THE FIRST CASE THE CAPABILITY ASSOCIATED WITH THE FIRST OCCURRENCE FOUND DURING BACKWARDS SCAN IS RETURNED AS THE RESULT. IN THE SECOND CASE, AN ERROR ( SEE BELOW ) OCCURS.

NOTE THAT MOST RECENTLY ADDED ENTRIES TO LOCAL DIRECTORY ARE SCANNED FIRST, FOLLOWED BY OLDER LOCAL DIRECTORY ENTRIES, FOLLOWED BY MOST RECENT GLOBAL DIRECTORY ENTRIES FOLLOWED BY OLDER GLOBAL DIRECTORY ENTRIES.

### D) IN THE TEXT INCLUDED IN THIS DOCUMENT ALL NAMES OF ENTRIES GIVEN WILL BE REPRESENTED IN DISPLAY CODE, LEFT JUSTIFIED, ZERO FILLED.

V'

## SUBPROCESS CONSTRUCTION

THE SUBPROCESS IS BUILT BY REFERENCE TO A DESCRIPTOR AT THE BEGINNING OF A FILE. ( SEE NEXT SECTION. ) THE TYPE IS CHECKED, THEN THE CLASS CODES FOR THE SUBPROCESS AND FOR ITS FATHER ARE OBTAINED FROM THE DIRECTORY, AS WELL AS THE INITIAL CALL OPERATION. NEXT A LOCAL C-LIST OF SPECIFIED SIZE IS CONSTRUCTED FOLLOWED BY A 1 LEVEL FILE, ( NO POINTER BLOCKS, ) OF SIZE SPECIFIED FOR THE SCRATCH FILE. THIS LOCAL C-LIST AND LOCAL SCRATCH FILE ARE ENTERED IN LOCAL DIRECTORY WITH THE SPECIFIED NAMES. A SUBPROCES OF SPECIFIED SHAPE, AND GIVEN CLASS CODES FOR ITSELF AND FATHER IS CONSTRUCTED. THE LOGICAL MAP ENTRIES ARE MADE, AND THE DESIRED C-LIST ENTRIES ARE FETCHED AND PLACED IN THE LOCAL C-LIST. FINALLY THE LOCAL C-LIST, SCRATCH FILE AND CLASS CODE ARE SAVED FOR LATER DESTRUCTION OF THE SUBPROCESS.



# V. SUBPROCESS DESCRIPTOR

THE GENERAL FORM IS THAT OF:

|                   |                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------|
| HEADER            | A BLOCK OF 13 WORDS                                                                                                    |
| MAP SPECIFIERS    | A SEQUENCE OF 3 WORD BLOCKS, EACH SPECIFYING A LOGICAL MAP ENTRY                                                       |
| -0                | MARKS END OF MAP SPECIFIERS                                                                                            |
| C-LIST SPECIFIERS | A SEQUENCE OF 1 WORD ITEMS TO NAME OBJECTS TO BE PLACED IN LOCAL C-LIST AT SAME RELATIVE POSITIONS AS IN THIS SEQUENCE |
| -0                | MARKS END OF C-LIST SPECIFIERS AND END OF DESCRIPTOR                                                                   |

## A) HEADER

|    |                                                     |
|----|-----------------------------------------------------|
| 0  | TYPE FIELD ( ALWAYS 1 )                             |
| 1  | ALWAYS ZERO                                         |
| 2  | NAME OF CLASS CODE FOR THIS SUBPROCESS              |
| 3  | NAME OF CLASS CODE FOR FATHER                       |
| 4  | NAME OF INITIAL CALL OPERATION                      |
| 5  | NAME TO BE USED IN LOCAL DIRECTORY FOR LOCAL C-LIST |
| 6  | NAME TO BE USED IN LOCAL DIRECTORY FOR SCRATCH FILE |
| 7  | LOGICAL MAP SIZE ( AN INTEGER )                     |
| 10 | COMPILED MAP SIZE ( AN INTEGER )                    |
| 11 | FIELD LENGTH ( AN INTEGER )                         |
| 12 | ENTRY POINT ( AN INTEGER )                          |
| 13 | C-LIST SIZE ( AN INTEGER )                          |
| 14 | SCRATCH FILE SIZE ( AN INTEGER )                    |

## B) MAP SPECIFIERS

|   |                                  |                    |
|---|----------------------------------|--------------------|
| 0 | FILE NAME                        |                    |
| 1 | LOGICAL MAP INDEX                | FILE ADDRESS       |
| 2 | CM ADDRESS                       | CM ADDRESS + COUNT |
|   | READ ONLY BIT ( 1 IF READ ONLY ) |                    |

## C) C-LIST SPECIFIERS

|   |                                                                       |
|---|-----------------------------------------------------------------------|
| 0 | OBJECT NAME OR ZERO                                                   |
|   | ( ZERO MEANS NO OBJECT WILL BE PLACED AT CORRESPONDING C LIST INDEX ) |



V. *2* MAKING A SYSTEM TAPE

A SUBPROCESS WILL EXIST ( NAME NOT YET DETERMINED ) WHICH IS CALLED WITH A FILE AS PARAMETER.

THAT PARAMETER FILE IS TO HAVE THE FORM OF A LIST OF 2 WORD BEAD DIRECTORY NAMES IN LEFT JUSTIFIED DISPLAY CODE.

THE FILES SO NAMED WILL BE TREATED AS FOLLOWS:

THE FIRST WILL BE TREATED AS PPU, CODE

THE SECOND WILL BE TREATED AS CPU, CODE

THE THIRD WILL BECOME THE BUILDER USED IN ALL PROCESSES AND IPROC. THIS WILL BE BUILDER, CODE.

THE FOURTH FILE WILL BE THE NAME FILE FOR SYSTEM INITIALIZATION ( SEE II )

SUBSEQUENT FILES WILL BE READ IN UNDER CONTROL OF BUILDER

THE END OF THIS LIST IS MARKED BY A ZERO WORD

V ) BUILDER, CODE

A) THIS FILE IS THE CODE FILE FOR ALL BUILDERS IN THE SYSTEM. IT HAS 3 USES.

- 1) SYSTEM INITIALIZATION AS THE BUILDER UNDER IPROC
- 2) A SINGLE USER OR CONTROL PROCESS INITIALIZATION
- 3) CONSTRUCTION OF ANOTHER PROCESS

B) ON A PARTICULAR CALL, THE ALGORITHM FOR DETERMINING THE PURPOSE OF THE CALL IS AS FOLLOWS:

- 1) IF THE CELL \*AGAIN\* IS NON ZERO, THIS IS A CALL TO CONSTRUCT ANOTHER PROCESS.
- 2) IF NOT, THE SCRATCH AREA IS INITIALIZED, AND A MAP ENTRY IS MADE FOR THE GLOBAL DIRECTORY. THE GLOBAL DIRECTORY IS CHECKED TO SEE IF ITS CONTROL WORD IS ZERO. IF SO, THIS A SYSTEM INITIALIZATION CASE AND A LOCAL DIRECTORY IS CREATED AND INITIALIZED ETC. IF NOT, THIS IS A PROCESS INITIALIZATION CASE AND A MAP ENTRY IS MADE FOR THE LOCAL DIRECTORY PASSED, ETC.



## I. TEST PACKAGE

### A) XIPROC,S

THIS SUBPROCESS ACCEPTS ONE TWO PART PARAMETER, A BEAD FILE NAME FOR A FILE. THIS FILE IS TO BE EXACTLY LIKE THE PARAMETER FILE FOR CONSTRUCTING A SYSTEM TAPE. THE 1ST 4 WORDS ARE IGNORED, ( 2 FILES ), THE 3RD FILE SHOULD BE #BUILDER, CODE# AND IS USED TO CONSTRUCT THE BUILDER. SUBSEQUENT FILES ARE TREATED IN SEQUENCE AS DESCRIBED UNDER SYSTEM INITIALIZATION PHASE. SEE II.

### B) XROOT,S

THIS SUBPROCESS IS USED TO SIMULATE THE ROOT OF A USER OR CONTROL PROCESS. IT ACCEPTS 1 ONE-PART PARAMETER AND TREATS THAT AS THE NAME PASSED IN THE NEW BUILDER SCRATCH FILE. ( THE NAME OF A FILE OF NAMES OF SUBPROCESS DESCRIPTOR FILE. ( THE NAME OF A FILE OF NAMES OF SUBPROCESS DESCRIPTORS ) THERE IS CURRENTLY NO WAY TO PRE-INITIALIZE THE LOCAL DIRECTORY. IF THIS IS NEEDED, SEE ME. IT PROBABLY WILL BE NEEDED EVENTUALLY.

### C) ADDITIONAL INFORMATION FOR XIPROC,S AND XROOT,S

- 1) THE GLOBAL DIRECTORY IS A FILE AND CLIST, IN BEAD DIRECTORY UNDER THE NAMES #GLBLD#, USER AND #GLBLC#, USER WHERE USER IS THE CURRENT USER NAME.
- 2) THE ITEM UNDER NAME #ROOT# IN THE LOCAL DIRECTORY IS THE CLASS CODE OF XIPROC OR XROOT RESPECTIVELY
- 3) THE BEAD CALL OPERATION IS AVAILABLE IN LOCAL DIRECTORY ( FOR TEST PURPOSES ONLY ) UNDER THE NAME #BEAD# . IT IS NOT MADE AVAILABLE DURING REAL SYSTEM INITIALIZATION OR IN PROCESSES CONSTRUCTED BY A CALL ON THE BUILDER AT ANY TIME. ONLY AVAILABLE UNDER XIPROC OR XROOT
- 4) SOURCE FILES ARE:

XIPROC,SOURCE  
XROOT,SOURCE  
BUILDER,SOURCE

### D) DIAGNOSTICS

NONEV

- 1) IF PRINTS OUT ..STOP THIS MEANS THAT BUILDER HAS DETECTED AN ERROR BY ITSELF. THE ONLY WAY TO DIAGNOSE THE ERROR IS TO DETERMINE WHERE IT OCCURED AND EXAMINE LISTINGS TO SEE WHAT IT WAS DOING. THE EXCHANGE PACK WILL BE STORED AT REGS IN BUILDER. ( CURRENTLY AT 1470, BUT CHECK CURRENT LISTINGS. ) BY CONTAINS THE ADDRESS

FOLLOWING THE INSTRUCTION WHERE THE ERROR WAS  
DETECTED. IF DETECTED AT AN XJ, THEN AN FRETURN  
OCCURED.

- 2) IF GET AN ERROR MESSAGE FROM THE BEAD, THEN USE  
SIMILAR PROCEDURES AS IN 1) EXCEPT THAT THE XJ PACK  
IS AT 32B IN THE BEAD.