SYSTEM AND SYSTEM PART USER PROCESS INITIALIZATION

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.

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.

Each part of the directory can be thought of as a seq of pairs. Each pair is a 60 bit word (the name) and 1 capability (the tag.) When ever 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 form 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 occurring at the beginning of a file and is described below.
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 THE OPERATIONS CONSTRUCTED BY INITIAL, EACH IS GIVEN THE NAME I 'NAME', WHERE 'NAME' IS THE SAME AS APPEARS IN C 'NAME' IN OPNAMES, XTEXT.
THE INITIAL VERSION OF THIS LIST WAS COPIED FROM KARL'S IPROC AND MAY BE INCOMPLETE.

B) THE MASTER C-LIST (IPROC'S C-LIST) AS I_MASTER.

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 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 FIELD IS NOW SCANNED STARTING WITH THE 1ST WORD AFTER THE 1ST ZERO WORD.


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

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

1) THE PROCESS IS CREATED WITH A SMALL ROOT SUBPROCESS CONTAINING
   A NAME, CAPABILITIES FOR THE DIRECTORY, TTY OBJECTS FOR
   PROCESSES CREATED FOR TTYS.

2) A SUBPROCESS IS CREATED (CALLED BUILDER) (WHAT FOLLOWS
   TAKES PLACE IN BUILDER.)

3) A LOCAL DIRECTORY IS INITIALIZED TO CONTAIN:
   A TTY FILE, REQUEST EVENT CHANNEL AND RESPONSE EVENT CHANNEL
   WITH NAMES
   'TTYFILE', 'ITYREQ', 'ITYRESP.'

4) A CLASS CODE FOR THE ROOT OF THE PROCESS, WITH NAME 'ROOT'.

5) 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.

6) 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.]

7) THE ROOT WILL NOW COMMIT SUICIDE.
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 SUBPROCESS 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.
SUBPROCESS DESCRIPTOR

THE GENERAL FORM IS THAT OF:

- HEADER
- MAP SPECIFIERS
- C-LIST SPECIFIERS
- MARKS END OF MAP SPECIFIERS
- 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 )
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 READ 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 FOR THE IPROC PROCESS
THE FOURTH FILE WILL BE THE NAME FILE FOR SYSTEM INITIALIZATION
(SEE II)

SUBSEQUENT FILES WILL BE READ IN UNLESS...

THE END OF THIS LIST IS MARKED BY A ZERO WORD
TEST PACKAGE

A) BUILDG,S

This subprocess accepts one two part parameter, a read name for a file, which is to be exactly as the parameter file for constructing a system tape. BUILDG,S ignores the 1st 6 words (1st 3 files) and begins with the name file and proceeds as described under system initialization phase. (See II)

B) BUILDR,S

This subprocess accepts one 1 part parameter, a name to be looked up in global directory. This subprocess acts exactly like that described in III, process initialization phase. The one parameter is treated as the name given to the root.

C) Additional information for BUILDG,S and BUILDR,S

1) The global directory is a file and c-list, in the read directory under GLBLD.USER and GLBLC.USER where user is current user name.

2) The item under the name root in local directory in both cases is the read class code.

3) The read call operation is available in local directory (for test purposes only) under the name 'read'. It is not made available during real system initialization or process construction.

D) Source files for test package

1) BUILDG,S

BUILDG,S SOURCE
BUILDG1,S SOURCE
BUILDG2,S SOURCE
BUILDG3,S SOURCE
BUILDG4,S SOURCE

Low core, data area for MC
Data area for SP
Main program
Subprocess builder, common subs
Top of core, globals

2) BUILDR,S

BUILDR,S SOURCE
BUILDR1,S SOURCE
BUILDR2,S SOURCE
BUILDR3,S SOURCE
BUILDR4,S SOURCE

Low core, data area for MP
Data area for SB
Main program
Subprocess builder, common subs
Top of core, globals
E) DIAGNOSTICS

NONE

1) IF PRINTS OUT "STOP THIS MEANS THAT BUILDLC'S OR
BUILDLR'S HAS DETECTED AN ERROR BY ITSELF. THE ONLY
WAY TO DIAGNOSE THE ERROR IS TO DETERMINE WHERE IT
OCCURRED AND EXAMINE LISTINGS TO SEE WHAT IT WAS
DOING. THE EXCHANGE PACK WILL BE STORED AT REGSVA
IN BUILDLR OR BUILDLC. (CURRENTLY AT 2123, BUT
CHECK CURRENT LISTINGS.) 97 CONTAINS THE ADDRESS
FOLLOWING THE INSTRUCTION WHERE THE ERROR WAS
DETECTED. IF DETECTED AT AN XJ, THEN AN FPETURN
OCCURED.

2) IF GET AN ERROR MESSAGE FROM THE BEAD, THEN USE
SIMILAR PROCEDURES AS IN 1) EXCEPT THAT THE XJ PACK
IS AT 323 IN THE BEAD.