GOOSY
Id.: GPS
Version: 1.0
Date: 19–JAN–1994
Revised: January, 19 1994

GSI Online Offline System

GOOSY PAW Server (GPS). Full Documentation

R.S. Mayer

January, 19 1994

GSI, Gesellschaft für Schwerionenforschung mbH
Postfach 11 05 52, Planckstraße 1, D-64220 Darmstadt
Tel. (0 6159) 71–0

Chapter 1

Preface

1.1 GOOSY Authors and Advisory Service

The authors of GOOSY and their main fields for advisory services are:

M. Richter GOOSY Data Management, VAX/VMS System Manager (Tel. 2394)

R. Barth GOOSY and PAW software (since 1995) (Tel. 2546)

H.G. Essel (GOOSY 1983-1993) Data Acquisition (Tel. 2491)

N. Kurz Data Acquisition (since 1992) (Tel. 2979)

W. Ott Data Acquisition (since 1994) (Tel. 2979)

People who have been involved in the development of GOOSY.

B. Dechant GOOSY software (1993-1095) (Tel. 2546)

R. S. Mayer Data Acquisition (1992-1995) (Tel. 2491)

R. Fritzsche Miscellanea (1989-1995) (Tel. 2419)

H. Grein Miscellanea (1984-1989)

T. Kroll Miscellanea, Printers (1984-1988)

R. Thomitzek Miscellanea, Printers, Terminals (1988-1989)

W. Kynast GIPSY preprocessor (1988)

W.F.J. Müller GOONET networking, Command interface (1984-1985)

H. Sohlbach J11, VME (1986-1989)

W. Spreng Display, Graphics (1984-1989)

K. Winkelmann GOOSY Data Elements, IBM (1984-1986)

1.2 Further GOOSY Manuals

The GOOSY system is described in the following manuals:

- GOOSY Introduction and Command Summary
- GOOSY Data Acquisition and Analysis
- GOOSY Data Management
- GOOSY Data Management Commands
- GOOSY Display
- GOOSY Hardware
- GOOSY DCL Procedures. GOOSY Error Recovery
- GOOSY Manual
- GOOSY Commands

Further manuals are available:

- GOOSY Buffer structures
- GOOSY PAW Server
- GOOSY LMD List Mode Data Generator
- SBS Single Branch System
- TCP-Package
- TRIGGER Bus
- VME Introduction
- OpenVMS Introduction

1.3 Intended Audience

This manual is written for GOOSY and PAW users. It assumes that the reader is familiar with most VAX-VMS concepts and commands and the environments (VMS, OpenVMS, UNIX-flavours) where PAW will be used. It provides all information necessary to use the GOOSY - PAW - Server. For VAX beginners the 'VMS Introduction' is recommended. For GOOSY beginners the 'GOOSY Introduction' is recommended.

Users of the PAW- and related software are kindfully asked to refer to the CERN software documentation. Our experience shows that updates in the software might result in different behavior. Please inform yourself about the current recommended version in your institute. Examples and templates for the use of the GPS client in a PAW environment are given here without any responsability of the GOOSY group and the user may **not** trust in regular updates.

1.4 Overview

- Section 2:
  GOOSY PAW Server: Introduction.
- Section 3:
  GPS GOOSY Server.
- Section 4:
  GPS Server Internals.
- Section 5:
  SBS Event Server.
- Section 6:
  GPS PAW Client.
- Section 7:
  GPS Client (standalone).
- Section 8:
  GPS Client Internals.

The author would be grateful for any critical comment or any suggestion about this manual.

1.5 GOOSY Copy Right

Chapter 2

Introduction

2.1 GOOSY - PAW - Server: Introduction

In order to allow online data acquisition control with other than GOOSY analysis software, the GPS tools have been developed. On the acquisition side

- The GPS-Server runs in the GOOSY Transport Manager environment.
- The SBS(Single Branch System) Event Server runs on the CVC (in preparation).

The GPS-Client(s) may run on one of the following computer platforms: VAX-VMS, AXP-OpenVMS, DEC-Ultrix, HP-UX, AIX (IBM). The client(s) may connect to the GOOSY server or SBS Event Server and request events. The client calls a user written analysis routine. Filters may be specified for the client to get selected events from the GOOSY server or SBS Event Server.

2.1.1 GPS GOOSY Server

The GPS Server can be started in a GOOSY environment with a running Transport Manager (see chapter 3).

The GOOSY PAW server may send event data from $TMR mailbox input to TCP/IP network clients. Serveral clients from the same or different hosts and systems may connect via TCP/IP and get data. The Clients may specify filter criteria, then only the selected event types are send to the client. Data input can be or from online data acquisition or from file input. When the server is started successfully, a port number is returned. This port number, together with the node name where the server runs, is required for the client to connect to the server.
The server tries to connect a port server and to communicate

- the name of the service, i.e. the full mailbox name used by the server as data input,
- the port number,
- the node name of the server and
- (a character string with information text).

Either the **service-name** (if not unique the **node::service-name**) when using the port server or **node** and **port** are required by the client.

2.1.2 SBS Event Server

The SBS Event Server (see chapter 5) is the equivalent part to the GPS server running on the CVC in the Single Branch System (SBS). The SBS Event Server get events from the data acquisition stream.

2.1.3 GPS PAW Client

This section (chapter 6) describes the full GPS client functionality.

The GPS-PAW-Client connect to a running Server (MGOOPS), send filter condition and request event data. After each event a user analysis routine is called (UANAL). The client connects to an existing server on a VMS node. To connect a client to a server, you need to specify the node and the port number. Alternatively and to simplificate the use, you may specify only the name of the server and port=0. (In case of ambiguity, i.e. several servers with the same name on different nodes, you must specify node::name). The client gets node and port from the port server and connects automatically.

Some hints are given how to build your own PAW when using the GPS client tool (see chapter 1.3). Until now, test have been done on VAX-VMS for the time being. When using the given command definition file

- an online help and
- an input menu

is available in the PAW environment.

2.1.4 GPS Client (standalone)

The GPS Client described in section 7 is a client test program. The functionality is the same as the GPS-PAW-Client routine, for the description refer also section 6. pc_client runs on VAX-VMS, AXP-OpenVMS and DEC-Ultrix. It will soon be available on AIX(IBM) and HP-UX.

2.1.5 Internals on GPS Server and Client

Internals on software modules used by the GPS Server and the GPS Client are available in the manual: **GOOSY PAW Server (GPS) Full Documentation** on VMS GOO$DOC:GM_GPS_FULL.PS. Please refer it when you intend to develope your own software using GPS.

Chapter 3

GPS GOOSY Server

MGOOPS

CALLING	GOOSY CRE PROC GPS $GPS
PURPOSE	GOOSY PAW server. Starts the GOOSY process $GPS

DESCRIPTION

CALLING	GOOSY CRE PROC GPS $GPS
FUNCTION	GOOSY PAW server may send event data from $TMR mailbox input to TCP/IP network clients. See GOOSY command START SERVER.

IMPLEMENTATION

Return type	BIN FIXED(31)
Status codes	Status of last command
File name	MGOOPS.PPL
Version	1.01
Author	H.G.Essel
Last Update	28-Jul-1993

INTERNALS

Utility	IO
Compile lib.	GOOINC.TLB
Home direct.	GOO$IO
Created	28-JUL-1993

Updates

Update	Date	Purpose
	10-Sep-93	Documentation (RSM)
	01-Dec-93	Delete name server entry (RSM)

MGOOPS_START

CALLING STS = MGOOPS_START(CV_envi,
L_channel,
I_clients,
I_sec)

PURPOSE Start GOOSY PAW server

DESCRIPTION

FUNCTION Starts GOOSY PAW server, which takes event data from the specified GOOSY mailbox. The mailbox is served by the GOOSY TransportMangeR either from an online data acquisition or from LMD-file input.

Several clients may connect to the server and receive event data. The number of allowed clients can be specified (max=20, def=10). For client interna see R.S.Mayer (Tel: -491). For the PAW implementation on VMS and UNIX please contact H.Dbbeling.

For further information see also the keywords MGOOPS, I$PS_SERVER and I$PS_PROC. Stop the server with the GOOSY command SHUT SERVER.

Command

Command keys START SERVER

START SERVER EnvName Channel Clients Loginter

PURPOSE Starts GOOSY PAW server, which takes event data from the specified GOOSY mailbox. Several clients may connect to the server and receive event data. See also MGOOPS.

PARAMETERS

EnvName Environment name (def=actual environment)

Channel Mailbox number 1,2 or 3 (def=1)

Clients Maximum number of clients allowed to connect (max=20, def=10)

Loginter	Interval [sec]write status information about the server and the connected clients to SLOG-file. (def=60)
Return type	BIN FIXED(31)
Status codes	Status of last command
File name	MGOOPS.PPL
Version	1.01
Author	H.G.Essel
Last Update	24-Aug-1993 (RSM)

UPDATES

Update	Date Purpose

MGOOPS_STOP

CALLING STS = MGOOPS_STOP()

PURPOSE Shut down GOOSY PAW server

DESCRIPTION

FUNCTION Shut down the GOOSY - PAW server, terminates the process $GPS. Delete name server entry.

Command

Command keys SHUT SERVER

SHUT SERVER

PURPOSE Shuts down the GOOSY PAW server, terminates the process $GPS

Return type BIN FIXED(31)

Status codes Status of last command

File name MGOOPS.PPL

Version 1.01

Author H.G.Essel

Last Update 24-Aug-1993 (RSM)

UPDATES

Updates Date Purpose

01-Dec-93 Delete name server entry (RSM)

Chapter 4

GPS Server Internals

I$PS_SERVER

```
l_sts = i$ps_server(c_mbxnam,i_maxclnts,i_sec)
```

PURPOSE GOOSY Transport Manager - PAW TCP/IP Server Connect to a GOOSY $TMR mailbox Setup connection for serveral clients. Get filters and send events to clients. The active server will be registrated in a Port Server.

ARGUMENTS

c_mbxnam (char *) Pointer to GOOSY mailbox name.

i_maxclnts (short) Maximum number of clients. (def=6, max=20)

i_sec (short) write status to logfile intervall [sec] (def=60).

Description

FUNCTION Server for event data from GOOSY mailbox. GOOSY Transport Manager - PAW TCP/IP Server.
Connect to a GOOSY $TMR mailbox from where it gets GOOSY buffers with event data.
Serveral clients from the same or different hosts and systems may connect via TCP/IP and get data.
The Clients may specify filter criteria, then only the selected event types are send to the client.
The active server will be registrated in a Port Server. The client(s) may connect by simply giving the name of the service (server_name) or in case of ambiguity the node_name::server_name. (See also MGOOPS)

Version 1.01

Author R.S. Mayer, R. Fritsche

Last Update 23-Aug-1993

Updates

Updates	Date	Purpose
	10-Sep-93	Documentation (RSM)
	01-Dec-93	Port Server connection included (RSM,RF)
	05-Dec-93	Extern. e_l_* to struct s_control (RSM)
	10-Dec-1993	Changed some var-types, TX sends endian and test_LW, check and swap only by RX. Modified buffer types: s_client, s_clntbuf,s_clntoutbuf,s_clnt_filter, gps_sc_def.
	02-Feb-1994	Include names and defines modified!!!
	03-Feb-1994	New STC-routines

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Comp. switch	on OpenVMS only : /STANDARD=VAXC
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB (only on VMS) SYS$SHARE:VAXCRTL.EXE/SH (only on VMS)
Created	23-Aug-1993
PROCEDURES	see I$PS_PROC The global control structure used in this package is s_control (see I$PS_PROC Structures)
STRUCTURES	see I$PS_PROC Structures
MACROS	see I$PS_PROC Macros

I$PS_PROC

none

PURPOSE GOOSY Transport Manager - PAW TCP/IP Server

Structures

s_bufhe GOOSY Buffer header.
module = GOOCINC(sa$bufhe)

s_client Client structure:

Referenced structures in
GOOCINC: s_iosb, s_itlst_1,
s_filter + s_flt_descr in module GOOCINC(s_filter)

s_clntbuf Output buffer structure that contains the data

s_clntoutbuf Output buffer structure contains control members and s_clntbuf for the output data

s_clnt_filter Data part that contains the filter (s_filter), the filter descriptor (s_flt_dscr) and some control and steering variables.

s_control General control structure for GOOSY-PAW-Server
Referenced structures in GOOCINC:
:BR.
s_iosb, s_client, s_evbuf, s_inbuf, s_itlst,
s_sockaddr
dsc$descriptor (SYS$LIBRARY:descrip.h)
e_p_control: external pointer on s_control

s_evbuf Buffer structure for spanned events.
module = GOOCINC(sa$evbuf)

s_event GOOSY event structure.
module = GOOCINC(sa$event)

s_filhe GOOSY file header.
module = GOOCINC(sa$filhe)

s_filter Filter for event selection.
module = GOOCINC(s_filter)

s_flt_descr Descriptor for filter.

s_inbuf Input buffer structure for GOOSY mailbox input (16 kByte). 16 buffers of this type are used

s_item Item list for VAX system service routines

s_itlst

s_itlst_1

s_iosb IO status block for SYS$QIO[W]requests

s_opc1 Structure for opcode bits in s_filter
module = GOOCINC(s_opc1)

s_pat s_pat1, s_pat2, s_pat3:
identities for filter pattern
module = GOOCINC(s_pat)

s_sockadr

s_ve10_1 GOOSY VME event structure.
module = GOOCINC(sa$ve10_1)

s_ves10_1 GOOSY VME subevent structure.
module = GOOCINC(sa$ves10_1)

Macros

gps_doqueue_mac macros $DOQUEUE_BEGIN(,,) and $DOQUEUE_END(,,) loop over all elements in a queue header created and filled with the S$FQ.... routines. See GOOCINC(gps_doqueue_mac) for explanation

m_pawser Contains Define Statements and Macros and declar. VAX System Services and Run Time Library and all used procedures for the PAW-Server.

gps_sc_def Contains Define Statements for MPAWSERVER and MPAWCLIENT.

gps_prtcl_def Declaration for i$ps_prtcl. See GOOINC(U$PRTCL)

Implementation

Version	1.01
Author	R. Fritsche, R.S.Mayer
Last Update	24-Jun-1993
Object libr.	GOOSHRLIB
Shar. Image	GOOIOSHR
Shar.Im.lib.	GOOSHR

Updates

Updates Date Purpose

01-Dec-1993	Delete from Port Server entry table, FQEND() instead of exit(1).
10-Dec-1993	Changed some var-types, TX sends endian and test_LW, check and swap only by RX. Modified buffer types: s_client, s_clntbuf,s_clntoutbuf,s_clnt_filter, gps_sc_def.
16-Dec-1993	Bugs removed (RSM)
02-Feb-1994	Include names and defines modified!!!
03-Feb-1994	New STC-routines

Internals

Utility	IO
Compile lib.	GOOINC
Comp. switch	on OpenVMS only : /STANDARD=VAXC
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB (only on VMS) SYS$SHARE:VAXCRTL.EXE/SH (only on VMS)
Created	24-Jun-1993

I$PS_ASCTIME

Module	I$PS_ASCTIME
CALLING	i$ps_asctime(char c_time[])
PURPOSE	Get system time and date.
ARGUMENTS	Character string c_time[25].
FUNCTION	Get VAX system time, converts to a string

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Impementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche, R.S. Mayer
Last Update	14-JUN 1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

I$PS_ACC_NXTCLNT_AST

Module	I$PS_ACC_NXTCLNT_AST
CALLING	sts = i$ps_acc_nxtclnt_ast(p_client)
PURPOSE	AST routine fired when a client connects
ARGUMENTS	p_client: Pointer to client structure s_client

FUNCTION	!!!!.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche, R.S. Mayer
Last Update	14-JUN 1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

CALLING	sts = i$ps_acc_nxtclnt_ast(p_client)
ARGUMENTS	p_client: Pointer to client structure s_client
FUNCTION	

I$PS_ACC_1STCLNT_AST

Module	I$PS_ACC_1STCLNT_AST
CALLING	sts = i$ps_acc_1stclnt_ast(p_client)
PURPOSE	AST routine fired when the first client connects
ARGUMENTS	p_client: Pointer to client structure s_client
FUNCTION	!!!!.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche, R.S. Mayer
Last Update	14-JUN 1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

CALLING sts = i$ps_acc_1stclnt_ast(p_client)

ARGUMENTS p_client: Pointer to client structure s_client

FUNCTION

I$PS_ASSIGNMBX

Module I$PS_ASSIGNMBX

CALLING sts = i$ps_assignmbx(c_mbxnam)

PURPOSE Assign the mailbox

ARGUMENTS c_mbxnam: (char *) Pointer to character string

FUNCTION !!!!.

Return type long (32 bit)

Status codes -

Initialize -

Include name -

Implementation

File name I$PS_PROC.C

Version 1.01

Author R.S. Mayer

Last Update 14-JUN-1993

Object libr. GOOSHRLIB

Internals

Utility IO

Compile lib. GOOCINC.TLB

Home direct. GOO$IO

Link option SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

CALLING	sts = i$ps_assignmbx(c_mbxnam)
PURPOSE	Assign the mailbox
ARGUMENTS	c_mbxnam: (char *) Pointer to character string
FUNCTION	

I$PS_BUFACKN_AST

Module	I$PS_BUFACKN_AST
CALLING	sts = i$ps_bufackn_ast(p_outbuf)
PURPOSE	AST routine receives acknoledge buffer from client
ARGUMENTS	p_outbuf: Pointer to structure s_clntoutbuf
FUNCTION	
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

CALLING	sts = i$ps_bufackn_ast(p_clntout)
PURPOSE	AST routine receives acknoledge buffer from client
ARGUMENTS	p_outbuf: Pointer to structure s_clntoutbuf
FUNCTION	

I$PS_BUFPRC_FORK

Module	I$PS_BUFPRC_FORK
CALLING	sts = i$ps_bufprc_fork(p)
PURPOSE	Process input buffer, unpack the events
ARGUMENTS	p: Address of pointer of an input buffer
FUNCTION	Process an input buffer that contains a GOOSY buffer, i.e. unpack also spanned events. Calls then filter routines etc.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

CALLING	sts = i$ps_bufprc_fork(p)
PURPOSE	Process input buffer, unpack the events
ARGUMENTS	p: Address of pointer of an input buffer
FUNCTION	Process an input buffer that contains a GOOSY buffer, i.e. unpack also spanned events. Calls then filter routines etc. This routine is handled by the fork queue manager

I$PS_BUILDEVT

Module	I$PS_BUILDEVT
CALLING	sts = i$ps_buildevt(p_client, p_event, p_current)
PURPOSE	Build a new event from p_client->lp_subevt_wrt[]. In p_client->lp_subevt_wrt[]the ptr's to the subevents are given.

ARGUMENTS

p_client	(struct s_client *) ptr to client struct.
p_event	(struct s_event *) ptr to event struct.
p_current	(char *) ptr to current position in output buffer.
FUNCTION	Build a new event from p_client->lp_subevt_wrt[]. In p_client->lp_subevt_wrt[]the ptr's to the subevents are given. In p_client->l_subevt_len[]the length of the subevents are specified.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_CHECKFLT

Module	I$PS_CHECKFLT
CALLING	sts = i$ps_checkflt(p_event)
PURPOSE	Loops over all clients and checks if the event is valid for individually specified filter conditions.
ARGUMENTS	p_event : Pointer to the original event
FUNCTION	Loops over all clients and checks if the event is valid for individually specified filter conditions and initiates the output of this event or parts of it to the output buffer of the client.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_CTRL_AST

Module	I$PS_CTRL_AST
CALLING	sts = i$ps_ctrl_ast(p_event)
PURPOSE	Timer AST routine for I$PS_CTRL_FORK
ARGUMENTS	dummy
FUNCTION	Timer AST routine for I$PS_CTRL_FORK. Queues the output of some server control values.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_CTRL_FORK

Module	I$PS_CTRL_FORK
CALLING	sts = i$ps_ctrl_fork(p)
PURPOSE	Output of some server control values
ARGUMENTS	p : Pointer on control constant **p = &e_p_control->e_l_null** Only output **p = &e_p_control->e_l_eins** Output and set up new timer AST
FUNCTION	Output of some server control values with or wo seting up a timer for the next output. This routine is handled by the fork queue manager
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_DEVDEF

Module	I$PS_DEVDEF
CALLING	i$ps_devdef(p)
PURPOSE	Evaluates device characteristics bits
ARGUMENTS	p: Pointer on union devdef
FUNCTION	Evaluates device characteristics bits. The used macro is defined in m_mbxfq.
Return type	-
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_FLUSH_AST

Module	I$PS_FLUSH_AST
CALLING	sts = i$ps_flush_ast(p_client)
PURPOSE	Timer AST routine for I$PS_FLUSH_FORK
ARGUMENTS	p : Address of pointer to client structure s_client
FUNCTION	Timer AST routine for I$PS_FLUSH_FORK.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_FLUSH_FORK

Module	I$PS_FLUSH_FORK
CALLING	sts = i$ps_flush_fork(p)
PURPOSE	Output buffer flushing
ARGUMENTS	p : Address of pointer to client structure s_client
FUNCTION	If no buffer has been written within a given time interval, the actual buffer or a new one (if available) will be sent to the client. This routine is handled by the fork queue manager
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_FETCH_OUTBUF

Module	I$PS_FETCH_OUTBUF
CALLING	sts = i$ps_fetch_outbuf(p_client)
PURPOSE	Fetch output buffer from client's free queue
ARGUMENTS	p_client: Pointer to client structure s_client
FUNCTION	Fetch a free output buffer, if available, from the individual queue of the client.
Return type	long (32 bit) 1: success
Status codes	**1** success **3** no buffer available
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_FLTDSCR_PRTCL

Module	I$PS_FLTDSCR_PRTCL
CALLING	sts = i$ps_fltdscr_PRTCL(p_clnt_filter)
PURPOSE	Read and check the event filter and construct the filter descriptor.
ARGUMENTS	p_clnt_filter: Pointer to structure s_clnt_filter
FUNCTION	Read and check the event filter and construct the filter descriptor. Output via U$PRTCL See also I$PC_FLTDSCR.
Return type	long (32 bit)
Status codes	bit 0: success bit 1: warning
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_GET_FILTER_FORK

Module	I$PS_GET_FILTER_FORK
CALLING	sts = i$ps_get_filter_fork(p)
PURPOSE	Setup the reading of the filter sent by the client
ARGUMENTS	p: Address of pointer to structure s_client
FUNCTION	Setup the reading of the filter sent by the client This routine is handled by the fork queue manager
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_GETMBXINF

Module	I$PS_GETMBXINF
CALLING	sts = i$ps_getmbxinf(c_mbxnam, if_print)
PURPOSE	Get information on mailbox
ARGUMENTS	
c_mbxnam	(char *) Pointer to mailbox name string
if_print	(short)
	1 Print info to SYS$OUT + LOG
	0 No output
FUNCTION	Get information on mailbox. Write information to structure s_control.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_GETMSG

Module	I$PS_GETMSG
CALLING	i$ps_getmsg(l_msgid,c_sysmsg,if_print)
PURPOSE	Get message for a system return code.
ARGUMENTS	
l_msgid	(long) return code
c_sysmsg	Pointer to message text output string
if_print	(short)

1	print message to screen
0	print not to screen

FUNCTION	Get message for a system return code and write it a 256 char string.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_INBUF_AST

Module	I$PS_INBUF_AST
CALLING	sts = i$ps_inbuf_ast(p_inbuf)
PURPOSE	AST routine for reading from mailbox
ARGUMENTS	p_inbuf: Pointer to structure s_inbuf

FUNCTION	AST routine for reading from mailbox
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_INBUF_FORK

Module	I$PS_INBUF_FORK
CALLING	sts = i$ps_inbuf_fork(p)
PURPOSE	Accept an input buffer from the mailbox
ARGUMENTS	p: Address of pointer to structure s_inbuf

p = -1	Flag: ignore stop-flag -> queue read.
p = 0	Flag: respect stop-flag -> queue read.
p > 0	Ptr to input buffer, queue read.

FUNCTION Accept an input buffer from the mailbox. Queues processing of this buffer (I$PS_BUFPRC_FORK) and sets up a new read on the mailbox This routine is handled by the fork queue manager

Return type long (32 bit)

Status codes -

Initialize -

Include name -

Implementation

File name I$PS_PROC.C

Version 1.01

Author R.S. Mayer

Last Update 14-JUN-1993

Object libr. GOOSHRLIB

Internals

Utility IO

Compile lib. GOOCINC.TLB

Home direct. GOO$IO

Link option SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates Date Purpose

14-Jan-1994	bug removed, modified (RSM)
18-Jan-1994	bug removed p<0,p=0, document. (RSM)

Description

I$PS_INIT_CLIENT_FORK

Module	I$PS_INIT_CLIENT_FORK
CALLING	sts = i$ps_init_client_fork(p)
PURPOSE	
ARGUMENTS	p:
FUNCTION	This routine is handled by the fork queue manager
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_INITNEXTCLIENT

Module	I$PS_INITNEXTCLIENT
CALLING	sts = i$ps_initnextclient(p)
PURPOSE	
ARGUMENTS	p:
FUNCTION	
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_INIT_SERVER

Module	I$PS_INIT_SERVER
CALLING	sts = i$ps_init_server(p)
PURPOSE	
ARGUMENTS	p:
FUNCTION	
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_MRKCLNTDEL

Module	I$PS_MRKCLNTDEL
CALLING	sts = i$ps_mrkclntdel(p_client)
PURPOSE	Mark client invalid, queue deletion of client.
ARGUMENTS	p_client: Pointer to client structure s_client
FUNCTION	Mark client invalid, close connection to client and queue deletion of client. Set i$ps_remclnt_fork(p) in the fork queue.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date	Purpose
	21-Jan-1994	Chgd. handl. not ackn. buf. (RSM)

Description

I$PS_PRTCL

Module	I$PS_PRTCL
CALLING	sts = i$ps_prtcl(p_string, b_mode)
PURPOSE	C function for U$PRTCL
ARGUMENTS	
p_string	Pointer to character string
b_mode	(long unsigned) Output mode
FUNCTION	See U$PRTCL
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_READ_CLIENT

Module	I$PS_READ_CLIENT
CALLING	sts = i$ps_read_client(p)
PURPOSE	
ARGUMENTS	p:
FUNCTION	
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_READ_FILTER_AST

Module	I$PS_READ_FILTER_AST
CALLING	sts = i$ps_read_filter_ast(p_client)
PURPOSE	AST routine for incomming filter data
ARGUMENTS	p_client: Pointer to client structure s_client
FUNCTION	AST routine for incomming filter data. Forks the i$ps_read_filter_fork routine.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_READ_FILTER_FORK

Module	I$PS_READ_FILTER_FORK
CALLING	sts = i$ps_read_filter_fork(p)
PURPOSE	Read the filter data, sets up the client.
ARGUMENTS	p: Address of pointer to clientstructure s_client
FUNCTION	Read the filter data, sets up the client, updates the control structure. Sends a first information buffer to the client and sets up the first buffer flushing timer AST. This routine is handled by the fork queue manager
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	23-AUG-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose	
	21-Jan-1994	(if_err) i$ps_mrkclntdel (RSM)

Description

I$PS_REMCLNT_FORK

Module	I$PS_REMCLNT_FORK
CALLING	sts = i$ps_remclnt_fork(p)
PURPOSE	Remove the client from the active queue.
ARGUMENTS	p: Address of pointer to client structure s_client
FUNCTION	Remove the client from the active queue. Puts the output buffers and the client structure in free queues. Updates control structure. This routine is handled by the fork queue manager
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer

Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_SENDBUF

Module	I$PS_SENDBUF
CALLING	sts = i$ps_sendbuf(p_client, p_buffer, l_buflen)
PURPOSE	Sends an output buffer to the client.
ARGUMENTS	
p_client	Pointer to client structure
p_buffer	Pointer to output buffer
l_buflen	(long) length of data (byte) in the data field s_clntoutbuf.c_buffer[]
FUNCTION	Sends an output buffer to the client. In all cases, the first 512 bytes are send. This includes the output buffer header and a part of the data. If required, the rest is sent with variable length in a second write operation.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_STRTOUPPER

Module	I$PS_STRTOUPPER
CALLING	sts = i$ps_strtoupper(u, l)
PURPOSE	Converts a '?0' terminated string to upper case.
ARGUMENTS	
u	Pointer to upper case string (result)
l	Pointer to lower case string (argument)
FUNCTION	Converts a '?0' terminated string to upper case.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_TYPFLT_PRTCL

Module	I$PS_TYPFLT_PRTCL
CALLING	sts = i$ps_typflt_prtcl(p_clnt_filter)
PURPOSE	Type the filter conditions.
ARGUMENTS	p_clnt_filter: Pointer to s_clnt_filter
FUNCTION	Type the filter conditions. Output via U$PRTCL. See also I$PC_TYPFLT.
Return type	long (32 bit) 1: success 0: fault
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date	Purpose
	16-Dec-1993	Debug: U$M_PRTSLOG repl. by U$M_PRTT
	12-Jan-1994	Bug removed in subcrate,control (RSM)

Description

I$PS_WRITE_CLIENT_AST

Module	I$PS_WRITE_CLIENT_AST
CALLING	sts = i$ps_write_client_ast(p_clnt_filter)
PURPOSE	AST routine for I$PS_WRITE_CLIENT.
ARGUMENTS	p_clnt_filter: Pointer to s_clnt_filter
FUNCTION	AST routine for I$PS_WRITE_CLIENT. If an error occurs, the client will be removed.
Return type	long (32 bit)
Status codes	-

Initialize -

Include name -

Implementation

File name I$PS_PROC.C

Version 1.01

Author R.S. Mayer

Last Update 14-JUN-1993

Object libr. GOOSHRLIB

Internals

Utility IO

Compile lib. GOOCINC.TLB

Home direct. GOO$IO

Link option SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates Date Purpose

Description

I$PS_WRITE_CLIENT

Module I$PS_WRITE_CLIENT

CALLING

```
sts = i$ps_write_client(p_client,
                        p_buffer,
                        l_buflen)
```

PURPOSE

ARGUMENTS

p_client Pointer to client structure

p_buffer Address of the first byte

l_buflen	Lenght in byte

FUNCTION

Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R. Fritsche
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date	Purpose
	26-Jan-1994	Bug in length of swap-field rem. (RSM)

Description

I$PS_WRT_EVT

Module	I$PS_WRT_EVT
CALLING	sts = i$ps_wrt_evt(p_client, p_event, if_bldevt)
PURPOSE	Copy event or parts of it to the output buffer.
ARGUMENTS	
p_client	Pointer to client structure s_client
p_event	Pointer to event
if_bldevt	(short) Flag

0	copy whole event
1	copy subevents

FUNCTION	Copy event or parts of it to the output buffer. Checks if there is enough memory in output buffer, otherwise takes a new output buffer if available if_bldevt=1 only valid for event-type=10 (see also I$PS_BUILDEVT)
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date Purpose

Description

I$PS_SHUTSERVER

Module	I$PS_SHUTSERVER
CALLING	sts = i$ps_shutserver()
PURPOSE	Routine to execute before shutdown server. Notify each active client of the shutdown by sending a message buffer.
ARGUMENTS	-
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Description

FUNCTION	Routine to execute before shutdown server. Notify each active client of the shutdown by sending a message buffer. The actual statistics are copied to the buffer, but there might be pending buffers with events allready accumulated in the statistics. So the numbers given may not be exactly the number of received events.

Implementation

File name	I$PS_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE

Updates

Updates	Date	Purpose
	05-Dec-1993	Now called by I$PS_TPS_DELSERV

I$PS_TPS_DELSERV

Module	I$PS_TPS_DELSERV()
CALLING	sts = i$ps_tps_delserv()
PURPOSE	Delete Server in the Port Server entry table.
ARGUMENTS	void
FUNCTION	Delete Server in the Port Server entry table.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Internals

Utility	IO
Compile lib.	GOOCINC.TLB
Home direct.	GOO$IO
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE
Created	01-Dec-1993

Implemetation

File name	I$PS_SERVER.C
Version	1.01
Author	R.S. Mayer
Last Update	01-DEC-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

CALLING	sts = i$ps_tps_delserv(p_client)
ARGUMENTS	p_client: Pointer to client structure s_client
FUNCTION	

Chapter 5

SBS Event Server

5.1 SBS Event Server

The connection and the event request is identical to the GOOSY GPS Server (see 3).

The SBS Event Server is available since May-94 and documented in the manual: **Single Branch System** on VMS GOO$DOC:GM_SBS.PS.

Chapter 6

GPS PAW Client

F_CLIPAW

```
f_clipaw(pc_hostn,pl_portn,pl_nbevt,pc_filtr,
         pl_sampl,pl_echo,pc_tyevt,pl_bflsh)
```

PURPOSE Client to connect GOOSY - PAW - Server or SBS - Event - Server. (See also HELP MGOOPS and documentation m_event_serv)

ARGUMENTS (see also Description Function)

pc_hostn (char *) Node_name

pl_portn (long *) Port_number > 0

or by using the port server

pc_hostn ptr to Server_name or Node_name::Server_name

pl_portn ptr to Port number <= 0

pl_nbevt (long *) Number of events

>0	Requ number of events from the server
=0	Client will be started with the given parameters but no connection established
-1	Request an unlimited number of events

pc_filtr (char *) Filter option or File name

A	Select all events.
F	Filter condition will be prompted.
file	File with filter specifications.

pl_sampl (long *) Reduction rate

0,1	Get every event when filter match.
n>0	Every n'th event when filter match.

pl_echo (long *) Echo rate

	>0	Notify after every n'th event.
	0	Disable echo.
	n<0	Notify every n percent of requ. events

pc_tyevt (char *) Display option

X	Switch off analysis routine UANAL
N	No display
H	Display only event (and subevt) Header
D	Display evt/sev Header and Data

pl_bflsh (long *) Buffer flushing time (timeout = *10)

Description

FUNCTION Connect to a running Server (GOOSY: MGOOPS or SBS: m_event_serv), send filter condition and request event data. After each event a user analysis routine is called (UANAL). The client connects to an existing server

GOOSY-GPS	GOOSY GPS Server on a VAX-[Open]VMS node. The server can be started from a GOOSY environment with a running transport-manager ($TMR) process: G>CREATE PROCESS GPS $GPS G>START SERVER
SBS-Event-Server	m_event_serv on SBS (Single Branch System)

Description (See VAX/VMS Help items MGOOPS, I$PS_SERVER and I$PS_PROC for further information). The starting server returns his port number and automatically reports his node, portnumber and name to a port server. Otherwise the Portnumber of the already running server must be known.

To connect a client to a server, you need to specify the node and the port number. Alternatively and to simplificate the use, you may specify only the name of the server and port=0. (In case of ambiguity, i.e. several servers with the same name on different nodes, you must specify node::name). The client gets node and port from the port server and connects automatically.

The client requests a number of events (or a continous stream) from the server. A strong selection on the events may be applied by specifiing filter conditions.

Samples of events that match the filter may be taken.

The reception of events may be echoed.

The display of the event (and subevent) header or the full data content may be selected. With the same parameter the user analysis UANAL may be switched on/off.

The server flushes the buffers to the client in a selectable time intervall. Even when the server gets no events from his input, or none of the events match the filter, the client gets a information buffer regularly.

Statistical information about the server and the client, i.e. read/written bytes, processed buffers, processed events, filter matching events etc. are containd in every buffer sent by the server.
CTRL_g invoces the call of a user modifiable (FORTRAN) subroutine UCLINFO that e.g. displays these information. (Response time on CTRL_g <= buffer_flushing_time)

CTRL_a terminates the client. (Response see above)

For every incomming event, the user written (FORTRAN) analysis routine UANAL is called. UANAL may also be skipped.

Filter_description

Filter criteria for event selection Filters may be applied on the event (i.e. event header in the case of event type=10) and/or on subevents (if there are).

Several filter may be defined with logical conditions between them. Filter specifications may be grouped in so called blocks with logical conditions between them. Each block applies on a different region of the event (e.g. 1st block on event header, trigger number etc., 2nd block on subevent a etc. ...). In case of a subevent, one must unambigiously define in the first entry of the filter block on which subevent the block shall be applied (by giving e.g. the processor id).

Output selection The server may send the whole event that has fullfilled the filter criteria or only parts of it. This selection is independent of the filter. E.g. in the filter, the trigger number (event) and a pattern in the subevent b is checked, but only the subevent a and c will be send to the client. The output selection is the first entry of complete filter.

Detailed filter description In order to maintain software performance, all filter specifications (see topic) are internally translated into bit mask and offset. The mask and offset may be word or longword aligned. This makes necessary that some rules for the filter definitions have to be respected:

The required order is:

1. output selection
2. filter selection
   a) event specific filter
   b) subevent specific filter

Negligible disregard of these rules are automatically corrected and result in a warning message.

A complete filter consists of

1. block with the output selection and
2. one or more blocks with filter criteria.

In some cases, you need only one block for output selection and filter criteria (see topic examples).

Defining a block:
File input and interactive input are identical!

1. lenght of block, i.e. number of filter entries
2. filter entry(ies)

The filter specification consists of 9 entries
(see filter_specification).

1: Select filter or output selection for
event (1) or subevent (0).
2: Select filter specification (1)
else (0).
3: Select output specification (1)
else (0).
4: Operation code: see Filter_specification
ALL,IDENT,ANY,INCL,EXCL,LT,GE:
filter result = mask opcode object
(the 'object' is the word or longword
at the position 'offset' in the
event data)
5: Logical link between filter

specifications in a filter block:
0: OR 1: AND
6: Logical link between filter blocks:
0: OR 1: AND
7: Filter specification:
0 : Take all
1 : trigger
2 : pattern and offset
4 : type
8 : subtype
12: subtype & type *)
16: procid
32: contr & subcrate *)
48: contr & subcrate & procid *)
*) byte/word sequ. from left to right
8: Mask (=bit pattern)
9: Offset

Filter_specification

Filter specification

1.evtsev 2.selflt 3.selwrt 4.opc 5.lnkf1 6.lnkf2
7.fltspec 8.mask 9.offs

1. Select event/subevent

1	event
0	subevent

2. Select filter

0	off
1	on

3. Select write

0	off
1	on

4. Object code [Res = object opcode mask]

0	!! (ALL)
1	== (IDENT) [object == mask]

2	&& (ANY) [object & mask]
3	&= (INCL) [(object & mask) == object]
4	^ = (EXCL) [(object & mask) == mask]
5	< (LT) [object < mask]
6	>= (GE) [object >= mask]

5. Logical link between filters in a filter block

0	OR
1	AND

6. Logical link between filter blocks

0	OR
1	AND

7. Filter specification and validity

0	Take all
1	trigger
2	pattern and offset
4	type
8	subtype
12	subtype & type *)
16	procid
32	contr & subcrate *)
48	contr & subcrate & procid *)
*)	byte/word sequ. from left to right
0 - 12	valid for events
2 - 48	valid for subevents

8. Mask (=bit pattern)
decimal or hex (0x.....) enter here the
Word or LongWord (see 9.) req. value for 7.

9. Offset
decimal or hex (but like 8.)
enter here the required
value for 7.(2)
otherwise 0 (will be set automatically)

Definition (see also 8.):
LW: offset >= 0 index on event or subevent
(0: 1st LW, ..., etc.)
W: offset < 0 index on event or subevent
(-1: 2nd W, ..., etc.)

Filter_examples

Filter Examples for interactive or file input

! and /* are allowed comment declarations!

1.a) Output: whole event.
Filter: Take all events with trigger >= 3

```
!
! output selection
1 ! block with 1 filter
1 0 1 0 0 0 0 0 0
! filter selection
1 ! block with 1 filter
1 1 0 6 0 0 1 3 0
```

1.b) is identical with 1.a)

```
!
1 ! block with 1 filter
1 1 1 6 0 0 1 3 0
```

2.a) Output: whole event.
Filter: Take all events with
trigger = 3 OR
trigger = 7
AND
the first three bits set
in the 15th LongWord of

```
                    subevent (processor id=20)
1 ! output selection
1 0 1 0 0 0 0 0 0
2 ! event filter selection
1 1 0 1 0 0 1 3 0
1 1 0 1 0 0 1 7 0
2 ! subevent filter selection
0 1 0 1 1 1 16 20 0
0 1 0 4 1 1 2 7 15 ! mask and offset decimal
```

```
2.b) Filter like 2.a) but
       Output subevent (processor id=20)
          and subevent (processor id=30)
2
0 0 1 1 0 0 16 10 0
0 0 1 1 0 0 16 20 0
2 ! event filter selection
1 1 0 1 0 0 1 3 0
1 1 0 1 0 0 1 7 0
2 ! subevent filter selection
0 1 0 1 1 1 16 20 0
0 1 0 4 1 1 2 0x7 0xF ! mask and offset hexadec.
```

User_routines

These routines have to be provided and linked
together

UANAL

Module	UANAL.FOR
CALLING	UANAL(I4EVT, I2STS, I4LEN)
PURPOSE	User analysis routine. Here, histogramming etc. has to be done.
PARAMETERS	
I4EVT	INTEGER*4 I4EVT(0:I4LEN) event vector (Longwords)

I4LEN	INTEGER*4 I4LEN data length in LW
I2STS	INTEGER*2 I2STS Return status (0: success)

UCLINFO

Module	UCLINFO.FOR
CALLING	UCLINFO(I4BUF1, I4BUF2)
PURPOSE	User info routine. Will be executed after the first and the last data buffer sent from the server and for each buffer with a CTRL_g keyboard input. For details see UCLINFO.FOR template and GOOCINC:S_CLNTBUF.H for available info and statistics data.

PARAMETERS

I4BUF1	INTEGER*4 I4BUF1(0:*) and
I4BUF2	INTEGER*4 I4BUF2(0:*) ptrs to server and client info data

Implementation

PROCEDURES	see PC_PROC
STRUCTURES	see PC_PROC Structures
MACROS	see PC_PROC Macros
Return type	none
File name	PC_CLIPAW.C
Version	1.01
Author	R.S. Mayer
Last Update	03-May-1993

Internals

Utility

Module name	F_CLIPAW
File name	PC_CLIPAW.C
Home direct.	TOOL$SOURCE
Compile lib.	TOOL$LIB:PC_GPS.TLB
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE
Created	01-Sep-1993

Updates

Updates Date Purpose

15-Dec-1993	Input, output buffer structure changed s_clntbuf_swap obsolet (RSM)
10-Jan-1994	Documentation
02-Feb-1994	Include names and defines modified!!!
03-Feb-1994	New STC-routines
17-Feb-1994	Handling read error (RSM)
25-Feb-1994	UCLINFO reinserted! (RSM)
15-Mar-1994	Server endian, adapted to new stc, RSM
11-Apr-1994	struct tcpcomm no more extern. Timeout bug removed. New function: f_send_ackn. Sending ackn. buff rem. from function f_read_server. Ctrl_a,_c,_z now like last buffer. Ackn. only after full treatment /RSM : Some slight modifications /RSM

6.0.1 PAW environment

On TOOL$SOURCE (VAX) you may find the following files:

- PAWMAIN.FOR
- UACTKD.FOR
- UANAL.FOR
- UCLINFO.FOR
- UKDEF.CDF

Then proceed as follows:

- To get the command definition: $ KUIPC UKDEF.CDF UKDEF.FOR
- Edit your own analysis program from UANAL.FOR. (You(!) must care about the byte order of the platform where UANAL shall work)
- Compile all FORTRAN files: $ for PAWMAIN,UACTKD,UANAL,UKDEF,UCLINFO
- Compile: $ cc tool$source:pc_clipaw.c+tool$lib:pc_gps.tlb/lib
- Compile: $ cc tool$source:pc_proc.c+tool$lib:pc_gps.tlb/lib
- link all with the option: sys$library:ucx$ipc.olb/lib, sys$share:vaxcrtl.exe/share. On VMS for the time being

Remark: On Alpha-VMS compile the C-sources with /STANDARD=VAXC. The link option is not required.

For further questions how to install your PAW environment, please contact the Data Analysis Group at GSI.

Chapter 7

GPS Client (standalone)

PC_CLIENT

PC_CLIENT(argc,argv)

PURPOSE Standalone client for GOOSY - PAW - Server or SBS - Event - Server.
(See HELP MGOOPS and documentation m_event_serv .
For test purpose. For application, PAW etc. see F_CLIPAW

ARGUMENTS

argc	Number of arguments + 1
argv[1]	ptr to Node_name
argv[2]	ptr to Port number (>0)
or	using the port server
argv[1]	ptr to Server_name or Node_name::Server_name
argv[2]	ptr to Port number <= 0
argv[3]	ptr to Number of events
argv[4]	ptr to Filter option or File name
argv[5]	ptr to Reduction rate
argv[6]	ptr to Echo rate
argv[7]	ptr to Display option
argv[8]	ptr to Buffer flushing time

Description

FUNCTION Client for test purpose GOOSY - PAW - Server and SBS - Event - Server.
(for detailed information see F_CLIPAW)

Implementation

PROCEDURES	see PC_PROC
STRUCTURES	see PC_PROC Structures
MACROS	see PC_PROC Macros
Return type	none
File name	PC_CLIENT.C
Version	1.01
Author	R.S. Mayer
Last Update	11-Apr-1994

Internals

Utility	
File name	PC_CLIENT.C
Home direct.	TOOL$SOURCE
Compile lib.	TOOL$LIB:PC_GPS.TLB
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB (on VMS only) SYS$SHARE:VAXCRTL.EXE/SH (on VMS only)
Created	01-Sep-1993

Updates

Updates	Date	Purpose
	15-Dec-1993	Input, output buffer structure changed s_clntbuf_swap obsolet (RSM)
	20-Jan-1994	_AIX compiles wo error (RSM)
	21-Jan-1994	modification first buffer handl. (RSM)
	24-Jan-1994	modifications (RSM)
	02-Feb-1994	Include names and defines modified!!!
	03-Feb-1994	New STC-routines
	17-Feb-1994	Handling read error (RSM)

25-Feb-1994	UCLINFO reinserted! (RSM)
15-Mar-1994	Server endian, adapted to new stc, RSM
11-Apr-1994	struct tcpcomm no more extern. Timeout bug removed. New function: f_send_ackn. Sending ackn. buff rem. from function f_read_server. Ctrl_a,_c,_z now like last buffer. Ackn. only after full treatment /RSM : Some slight modifications /RSM

Chapter 8

GPS Client Internals

PC_PROC

none

PURPOSE Procedure for GOOSY Transport Manager - PAW TCP/IP Server (see F_CLIPAW, MGOOPS)
A copy is on GOO$IO:I$pc_proc.c used by MGOOANL

Structures

s_clntbuf	Client buffer structure for the data from server
s_clntoutbuf	Output buffer structure
s_clnt_filter	Filter structure to be sent to server
s_event	GOOSY event structure. module = sa_event.h
s_filter	Filter for event selection.
s_flt_descr	Descriptor for filter
s_pat1	Descriptor for filter pattern module = s_pat.h
s_pat2	Descriptor for filter pattern module = s_pat.h
s_pat3	Descriptor for filter pattern module = s_pat.h
s_opc1	Descriptor for filter opcode
s_iosb	I/O status block (only VMS and OpenVMS)
s_keyb	Structure for keyboard read (see ttystuff.h)
f_stccomm	Include for TCP/IP communication
s_usl	Union unsigned long and 2 unsigned words
s_ve10_1	GOOSY VME event structure. module = sa_ve10_1.h
s_ves10_1	GOOSY VME subevent structure. module = sa_ves10_1.h

s_vesfb	Fastbus structure. See GOOINC(SA$VESFB)
s_fbdata	Fastbus structure. See GOOINC(SA$FBDATA)
'struct'_swap	Structures as above, but byte swapped for portation to HPUX, AIX and other big endian comp.

Utility

ttystuff	System dependent keyboard readout. C-code!

Define

m_pawcli	Contains function prototypes
gps_sc_def	Contains Define Statements for GPS-SERVER, GPS-CLIENT and SBS-Monitor. !! Has allways (!) to be consistent !!
clnt_buf_def	definitions of buffer sizes for GPS-SERVER, GPS-CLIENT and SBS-Monitor. !! Has allways (!) to be consistent !!

Updates

Updates	Date	Purpose
	20-Jan-94	_AIX compiles wo error (RSM)
	02-Feb-94	Include names and defines modified!!!
	08-Apr-94	f_clnup accepts NULL pointer. Some printout shortened. /HE
	11-Apr-94	Seperate routine for buffer ackn. Sending ackn. buff rem. from function f_read_server. /RSM

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer, R. Fritzsche
Last Update	14-JUN-1993

Internals

Utility	
File name	PC_PROC.C
Home direct.	TOOL$SOURCE
Compile lib.	TOOL$LIB:PC_GPS.TLB
Link option	SYS$LIBRARY:UCX$IPC.OLB/LIB SYS$SHARE:VAXCRTL.EXE/SHARE
Created	01-Sept-1993

F_FLTDSCR

Module	F_FLTDSCR
CALLING	sts = f_fltdscr(p_clnt_filter)
PURPOSE	Read and check the event filter and construct the filter descriptor.
ARGUMENTS	p_clnt_filter: Pointer to structure s_clnt_filter
FUNCTION	Read and check the event filter and construct the filter descriptor. Output via "printf". See also I$PS_FLTDSC_PRTCL
Return type	long (32 bit)
Status codes	bit 0: success bit 1: warning
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

F_FLTINI

Module	F_FLTINI
CALLING	sts = f_fltini(p_clnt_filter)
PURPOSE	Prompts filter specification from the keyboard
ARGUMENTS	p_clnt_filter: Pointer to structure s_clnt_filter
FUNCTION	Prompts filter specification from the keyboard
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose	
	25-Jan-1994	problems with sscanf removed (RSM)

Description

F_FLTRD

Module	F_FLTRD
CALLING	sts = f_fltrd(p_clnt_filter, c_file)
PURPOSE	Reads filter specification from a file
ARGUMENTS	
p_clnt_filter	Pointer to structure s_clnt_filter
c_file	Pointer to file name string
FUNCTION	Opens the file and reads filter specification
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date	Purpose
	25-Jan-1994	problems with sscanf removed (RSM)

Description

F_TYPEVT

Module	F_TYPEVT
CALLING	sts = f_typevt(p)
PURPOSE	Types event header (evt-type =4, 10) and subevent header (evt-type: 10).
ARGUMENTS	p: Pointer to event
FUNCTION	Types event header (evt-type =4, 10) and subevent header (evt-type: 10). Output via "printf".
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

F_LSTEVT

Module	F_LSTEVT
CALLING	sts = f_lstevt(p)

PURPOSE	Types event header (evt-type =4, 10) and subevent header (evt-type: 10). List the event data.
ARGUMENTS	p: Pointer to event
FUNCTION	Types event header (evt-type =4, 10) and subevent header (evt-type: 10). List event data, module header etc. Output via "printf".
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

F_TYPFLT

Module	F_TYPFLT
CALLING	sts = f_typflt(p_clnt_filter)
PURPOSE	Type the filter conditions.
ARGUMENTS	p_clnt_filter: Pointer to s_clnt_filter
FUNCTION	Type the filter conditions. Output via "printf". See also I$PS_TYPFLT_PRTCL.
Return type	long (32 bit) 1: success 0: fault

Status codes -

Initialize -

Include name -

Implementation

File name PC_PROC.C

Version 1.01

Author R.S. Mayer

Last Update 14-JUN-1993

Object libr. GOOSHRLIB

Updates

Updates Date Purpose

12-Jan-1994 Bug removed in subcrate,control (RSM)

Description

F_UNPACK_BUF

Module F_UNPACK_BUF

CALLING sts = f_unpack_buf(p_clntbuf,
if_typevt,
l_evtecho)

PURPOSE Loops over all events and types evt- and sev-header

ARGUMENTS

p_clntbuf Pointer to structure s_clntbuf

if_typevt (short) flag for header & data display

1	no display
2	display event & subevt header
3	display event & subevt header & data

	<0	no call of UANAL analysis routine abs. value of if_typevt equal to 1-3
l_evtecho	Echo each l_evtecho event processed	
FUNCTION	Loops over all events and types evt- and sev- header. (See also F_TYPEVT, F_LSTEVT) UANAL in general a FORTRAN analysis routine: UANAL(I4EVT(),I2STS,I4LEN) I4EVT(0:I4LEN-1): integer*4 event data vector I4LEN : integer*4 length of vector I2STS : Return status 0=success	
Return type	long (32 bit)	
Status codes	-	
Initialize	-	
Include name	-	

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

F_READ_SERVER

Module	F_READ_SERVER
CALLING	sts = f_read_server(p_clntbuf, l_maxbyt, p_bytrd, l_timeout, i_chan)

PURPOSE	Read a buffer from the server
ARGUMENTS	
p_clntbuf	Pointer to structure s_clntbuf
l_maxbyt	(long) maximum buffer size
p_bytrd	Pointer to (long) Number of read bytes
l_timeout	(long) Timeout in seconds
i_chan	(int) channel number
FUNCTION	Read a buffer of the type s_clntbuf from the server.
Return type	long (32 bit)
Status codes	1: success 0: fault
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date	Purpose
	26-Jan-1994	Swap inserted (RSM)
	24-Feb-1994	Bug removed (RSM)

Description

F_SEND_ACKN

Module	F_SEND_ACKN
CALLING	sts = f_send_ackn(l_clnt_sts, i_chan)
PURPOSE	Send acknoledge buffer to the server
ARGUMENTS	
l_clnt_sts	Status. Status bits will be set in addition to the status bits set by f_read_server in s_ackn struct.
i_chan	(int) channel number
FUNCTION	Send the acknoledge buffer. Set additional bits in the status word, i.e. "last buffer" etc.
Return type	long (32 bit)
Status codes	1: success 0: fault
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	11-Apr-1994
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

F_STRTOUPPER

Module	F_STRTOUPPER
CALLING	sts = f_strtoupper(u, l)
PURPOSE	Converts a '?0' terminated string to upper case.
ARGUMENTS	
u	Pointer to upper case string (result)
l	Pointer to lower case string (argument)
FUNCTION	Converts a '?0' terminated string to upper case.
Return type	long (32 bit)
Status codes	-
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date Purpose

Description

F_CLNUP

Module	F_CLNUP
CALLING	f_clnup(v_mem, p_keyb)
PURPOSE	Cleanup allocated memory and dealloc devices
ARGUMENTS	
v_mem[ ]	(long) [0]:maxidx=n [1:n]:ptr to allocated memory
p_keyb	Pointer to s_keyb or NULL.
FUNCTION	Cleanup allocated memory and dealloc devices Calls free(v_mem[i]) and f_ttydass(p_keyb)
Return type	void
Status codes	-
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	14-JUN-1993
Object libr.	GOOSHRLIB

Updates

Updates	Date	Purpose
	14-Jan-94	prototype TTYSTUFF(f_ttydass()) (RSM)
	08-Apr-94	Accept NULL pointer for p_keyb. /HE

Description

F_SWAPLW

Module	F_SWAPLW
CALLING	sts = f_swaplw(p_source, l_len, p_dest)
PURPOSE	Long word byte swap.
ARGUMENTS	
p_source	pointer to source.
l_dest	length (in long words)
p_dest	pointer to destination or 0 if destination = source.
FUNCTION	Long word byte swap. Works on the source field if pp_dest points to value 0 or swaps from the source to the destination field. (Should be replaced by a fast assembler routine)
Return type	long (32 bit)
Status codes	bit 0: success
Initialize	-
Include name	-

Implementation

File name	PC_PROC.C
Version	1.01
Author	R.S. Mayer
Last Update	27-Jan-1994
Object libr.	?

Updates

Updates	Date Purpose

Description

Contents

1 Preface **1**
1.1 GOOSY Authors and Advisory Service . . . 1
1.2 Further GOOSY Manuals . . . 2
1.3 **Intended Audience** . . . 3
1.4 Overview . . . 3
1.5 GOOSY Copy Right . . . 3

2 Introduction **5**
2.1 **GOOSY - PAW - Server: Introduction** . . . 6
2.1.1 GPS GOOSY Server . . . 6
2.1.2 SBS Event Server . . . 6
2.1.3 GPS PAW Client . . . 7
2.1.4 GPS Client (standalone) . . . 7
2.1.5 Internals on GPS Server and Client . . . 7

3 GPS GOOSY Server **9**
MGOOPS . . . 10
MGOOPS_START . . . 11
MGOOPS_STOP . . . 13

4 GPS Server Internals **15**
I$PS_SERVER . . . 16
I$PS_PROC . . . 18

5 SBS Event Server **63**
5.1 SBS Event Server . . . 64

6 GPS PAW Client **65**
F_CLIPAW . . . 66
6.0.1 PAW environment . . . 76

7 GPS Client (standalone) **77**
PC_CLIENT 78

8 GPS Client Internals **81**
PC_PROC 82