YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SNA AĞ YÖNETİM SİSTEMİ

Bilgisayar Mühendisi Selviye Ferhatoğlu

F.B.E. Bilgisayar Bilimleri Mühendisliği Anabilim Dalında

hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Y.Doç. Dr. Tülay Tinli

İSTANBUL, 1996

İÇİNDEKİLER

ŞEKİL LİSTESİ

KISALTMALAR LİSTESİ

ACB : Access Control Block
API : Application Programming Interface
APPL : Application Logical Unit
CINIT : Control Initiate
CTERM : Control Terminate
DECNET : Digital Equipment Corporation Network
ECB : Event Control Block
EXLST : Exit List
EXRT : Exit Routine
LU : Logical Unit
NAU : Network Addresable Units
NCCF : Network Communication Control Facility
NCP : Network Control Program
NIB : Node Initialisation Block
OCIR : Operator Communication Interruption Routine
PLU : Primary Logical Unit
PU : Physical Unit
RPL : Request Parameter List
RQ : Request Unit
RSP : Response Unit
RU : Request/Response Unit
SCS : SNA Character String
SDLC : Synchronous Data Link Control
SDT : Start Data Traffic
SLU : Secondary Logical Unit
SNA : Systems Network Architecture
SPO : Secondary Program Operator
SSCP : Systems Services Control Point
VAX : Virtual Address Extension
VMS : Virtual Memory System
VSE/ESA : Virtual Storage Extended / Enterprise Systems Architecture
VTAM : Virtual Telecommunications Access Method

TEŞEKKÜR

Tez çalışmamın başlangıcından bitimine kadar geçen süre boyunca destek veren hocam Prof. M. Yahya Karslıgil'e ve hocam Y.Doç. Tülay Tinli'ye, çalışmamın her aşamasında desteklerini esirgemeyen tüm arkadaşlarıma teşekkür ederim.

ÖZET

Bu tezin amacı, kullanışlı bir SNA Ağ Yönetim Sistemini geliştirmekti. Bu yönetim sistemi, SNA ağ kaynaklarının durumunun gözlenmesi, kaynakların etkinleştirilmesi veya işlem-dışı durumuna getirilmesi gibi işlemlerin, yani ağ kaynaklarının denetiminin bir VAX/VMX sisteminden gerçekleşmesini sağlamaktadır.

Bu Ağ Yönetim Sistemi iki mödül içermektedir: NCCF (Network Communication Control Facility) ve Status Monitor (Durum İzleme). NCCF, ağ kaynaklarının denetimi için gerekli VTAM (Virtual Telecommunication Access Method) işletmen komutlarının girilmesi ve işletmen mesajlarının görüntülenmesini sağlayan ortamdır. Status Monitor ise, aynı çeşit ağ kaynaklarının durumlarını tek ekranda görüntüleyen ve bu kaynaklarla ilgili işlemlerin fonksiyon tuşları ile yapılmasına izin veren ortamdır.

Tez çalışmam üç uygulamadan oluşmaktadır. Uygulamalardan biri, IBM VSE/ESA işletim sistemi altında, S/370 Assembler dili ile geliştirdiğim, VTAM uygulamasıdır. Diğer iki uygulama ise, VAX VMS işletim sistemi altında, C programlama dilinde geliştirdiğim SNAServer ve Kullanıcı Arayüzü 'dür. Bu üç uygulama arasında veri alış-verişi gerçekleştirilmektedir. VTAM uygulaması ile SNAServer arasındaki veri aktarımı bir SNA LU-LU tip 0 oturumu üzerinden yapılmaktadır.

ABSTRACT

The aim of this thesis work was to develop an user-friendly SNA Network Management System that provides a means of monitoring and managing the SNA network resources from a VAX/VMS system.

This system includes two modules: NCCF (Network Communication Control Facility) and Status Monitor. NCCF is an environment from which the user can enter all the VTAM operator commands necessary for the control of the network resources and view the VTAM operator messages. The Status Monitor is an environment that displays the resources of the same type on a single screen and makes possible to perform the necessary operations on these resources only using the functions keys.

This thesis work is composed of three different applications. The first is a VTAM Application that was developped under the IBM VSE/ESA operating system, using the S/370 assembler language. The other two applications are the SNAServer and User Interface that were developed under the VAX VMS operating system, using the C programming language.

BÖLÜM 1

1.1 Genel Giriş

Bu tez çalışmanın amacı bir SNA Ağ Yönetim Sistemini gerçekleştirmekti. Geliştirdiğim sistem, SNA (Systems Network Architecture) ağ kaynaklarının izlenmesi ve denetimi için kullanışlı bir ortam sağlayan, çok kullanıcılı bir sistemdir.

SNA, IBM bilgisayar ürünleri arasında iletişimi sağlayan hiyerarşik bir ağ yapısıdır. SNA ağı fiziksel ve kavramsal kaynaklar içermektedir. SNA 'de bulunan fiziksel kaynaklar şunlardır: anasistemler, iletişim denetleyicileri, çevresel denetleyiciler, iş istasyonları ve iletişim hatları (LINE) 'dır. SNA' de üç çeşit kavramsal kaynak bulunmaktadır : SSCP (Systems Services Control Point - Sistem Servisleri Denetim Noktası), PU (Physical Unit - Fiziksel Birim) ve LU (Logical Unit - Kavramsal Birim). Bu ağ kaynakları kavramsal olarak Büyük Düğüm'lere (Major Node) gruplandırılmaktadır.
-PU, bulunduğu düğümün kaynaklarını ve ona bağlı çevresel aygıtları denetleyen bileşendir.
-LU'lar, uç-kullanıcıların SNA ağına erişmelerini ve aralarında veri iletişiminin gerçekleşmesini sağlarlar. Bir LU, bir iş istasyonunu, bir yazıcıyı veya bir uygulama programını temsil edebilir.
-SSCP, bir bilgisayar ağındaki kaynakları kontrol eder, oturumların kurulma sürecini denetler. SSCP anasistem düğümünlerinde bulunur ve VTAM (Virtual Telecommunications Access Method) erişim yöntemi programında gerçekleştirilir. VTAM, ağ kaynakların denetlenmesini, işletmen komutları ve işletmen mesajları vasıtasıyla sağlar.

Bir SNA Ağ Yönetim Sistemi, Büyük Düğüm'leri, kavramsal ağ kaynakları ve bu kaynaklar arasındaki oturumların (session) denetlenmesini sağlayan bir ortamdır. Bunu gerçekleştiren birçok ürün bulunmaktadır. Bunlardan ikisi Legend Netmaster ve IBM Netview'dür. Bu iki ürün bir çok modül içeren karmaşık Ağ Yönetim Sistem'leridir.

IBM'in geliştirdiği SNA Ağ Yönetim Sistemi Netview'dur. Netview dört ana modül içermektedir. Bunlar NCCF (Network Communication Control Facility), Hardware Monitor (Donanım İzleme), Session Monitor (Oturum İzleme) ve Status Monitor (Durum İzleme). İletişim operatörleri genellikle NCCF mödüllünü kullanmaktadırlar. Operatörlerin çalışmasından edindiğim izlenim, diğer mödülleri kolay anlaşılır ve kullanışlı bulmadıkları için tercih etmemeleridir. Aynı çeşit ağ kaynakları tek bir ekrandan izleme ve denetleme olanağını sağlayan kolay anlaşılır bir ortama ihtiyaç olduğunu anlayınca, önceleri Netview altında bir uygulama geliştirdim. Bu uygulamayı S/370 assembler dilinde yazdım ve Netview makrolarını kullanarak kullanıcı ekranı ile veri alış-verişi gerçekleştirdim. Uygulama ağdaki tüm hatların durumlarını görüntüleyen LINES ekranı, tüm PU'ları görüntüleyen

PUS ekranı ve tüm Büyük Düğüm`leri görüntüleyen MAJNOD ekranını içeriyordu. Fakat geliştirdiğim uygulama, Netview programını yavaşlattı ve IBM anasistemin işlemcisini gerektiğinden fazla meşgul etti. Bu nedenle, bu tez çalışmasında VTAM makrolarını doğrudan kullanarak bir VTAM uygulaması geliştirmeye karar verdim.

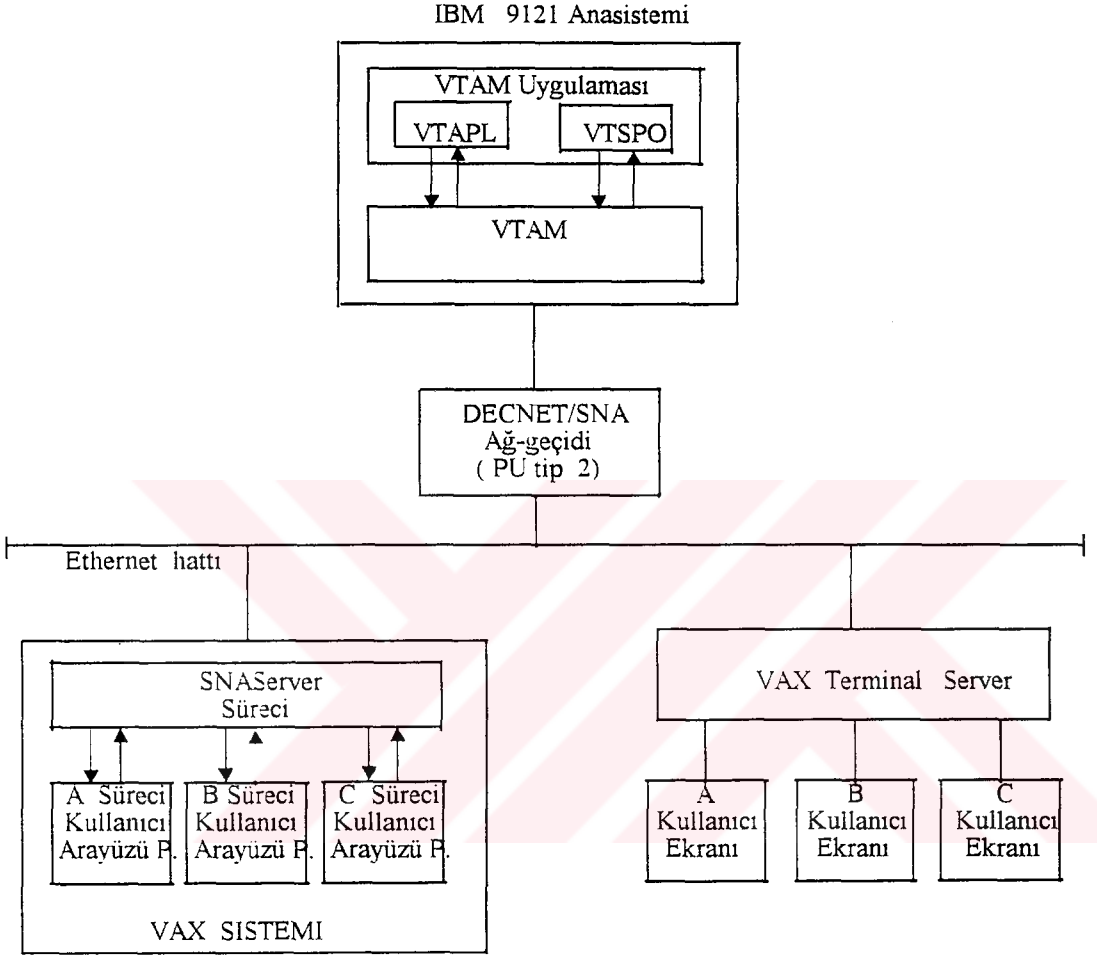
Geliştirdiğim SNA Ağ Yönetim Sistemi, NCCF ve Status Monitor mödüllerini içermektedir. NCCF mödülü VTAM işletmen komutlarının girilmesini sağlayan ortamdır. Ayrıca, bu ortamda işletmen komutlarına cevap olan VTAM işletmen mesajları görüntülenmektedir. NCCF'te bilinen VTAM komutları dışında CLIST komutları da kullanılabilir. CLIST'ler komut listeleridir ve birden fazla VTAM komutunun bir tek kelimenin girilmesi ile işlenmesini sağlarlar. Örneğin 'AALL` komutu SNA ağındaki tüm PU`ların etkin durumuna gelmesini sağlayan VTAM işletmen komutları içermektedir. Status Monitor mödülü üç ayrı görüntüleme ekranı içermektedir: MAJNOD, LINES ve PUS. MAJNOD ekranı, SNA ağında bulunan tüm Büyük Düğüm`lerin durumlarını görüntüler. LINES ekranı ağda bulunan tüm iletişim hatlarının durumlarını izleme ekranıdır. PUS ekranı ise tüm Fiziksel Birim`lerin durumlarını izleme ekranıdır. Bu ekranlar belli zaman aralıklarında yenilenerek ağ kaynaklarının son durumlarını gösterirler. Durum izleme ekranlarında bir ağ kaynağı seçilebilir ve fonksiyon tuşlarını kullanarak seçilen kaynakla ilgili şu işlemler gerçekleştirilebilir: kaynak hakkında ayrıntılı bilgi edinir, kaynak etkin veya işlem-dışı durumuna getirilebilir veya alt-kaynaklar üzerinden işlemler yapılabilir.

Yaptığım çalışma üç ayrı uygulamadan oluşuyor. Bunlardan ilki, IBM anasisteminde VSE/ESA işletim sistemi altında S/370 Assembler dilinde geliştirdiğim bir VTAM Uygulamasıdır. Tezin diğer iki uygulamasını ise, VAX sisteminde VMS işletim sistemi altında C programlama dilinde geliştirdim. Bunlar SNAServer ve Kullanıcı Arayüzü`dür.

Geliştirdiğim VTAM Uygulaması, SNA ağında iki Uygulama LU`su tarafından temsil ediliyor. Bu LU`ların ismi VTAPL ve VTSPO'dur. VTAPL ismindeki Uygulama LU`sunu, SNAServer ile veri alış-verişinde bulunması için tanımladım. VTSPO ismindeki Uygulama LU`su ise bir SPO'dur (Secondary Operator Program - İkincil Program İşletmeni). SPO, VTAM'a VTAM komutlarının iletilmesi ve bu komutların sonucu olan, VTAM mesajlarının elde edilmesini sağlar. SNAServer ile veri iletişimin gerçekleştirilmesi, VTAM'a işletmen komutlarının iletilmesi ve işletmen mesajlarının alınması için, VTAM Uygulamasında, VTAM makrolarını kullandım.

SNAServer, Şekil 1.1'de görülen DECnet/SNA ağ-geçidi altında tanımlı, bir LU tarafından temsil ediliyor. Bu LU'nun ismi LUSERV'dır. LUSERV ve VTAPL kavramsal birimleri arasında bir LU-LU oturumu kurulmaktadır. Bu oturumun kurulması, sonlandırılması ve kurulan oturum üzerinden veri

alış-verişinin sağlanması için, SNAServer uygulamasında DECnet/SNA API (Application Programming Interface) fonksiyonlarını kullandım.



Şekil 1.1 SNA Ağ Yönetim Sistemi'nin yapısı

VAX sisteminde bulunan VMS işletim sistemi yazılım uygulamalarının geliştirilmesi icin zengin bir ortam sağlamaktadır. VMS'te, her kullanıcı sisteme 'login' olduğu zaman bir kullanıcı süreci yaratılır. Bu tezde geliştirdiğim Kullanıcı Arayüzü, kullanıcı sürecinde çalıştırılmaktadır. Bu uygulama birden fazla kullanıcı tarafından kullanılabilir. Bu durumda VMS, her kullanıcı sürecinde uygulamanın ayrı bir görüntüsünü çalıştırır. SNAServer ise, SNAServer ismini taşıyan sürecin altında çalıştırılmaktadır. VMS'in sağladığı sistem fonksiyonlarını kullanarak, Kullanıcı Arayüzünün çalıştırıldığı süreç ve SNAServer süreci arasında veri iletişimini gerçekleştirdim.

Geliştirdiğim uygulamalar arasındaki veri akışı şu şekilde gerçekleşmektedir:
İzleme ekranlarının gerektirdiği veya NCCF ekranından kullanıcı tarafından girilen, VTAM işletmen komutları, kullanıcı sürecinden SNAServer sürecine aktarılır. SNAServer, kullanıcı sürecinden aldığı VTAM işletmen komutunu, VTAM Uygulamasına iletir. VTAM Uygulaması ise, bu komutun çalıştırılması için VTAM'a talepte bulunur ve komutun sonucu olan VTAM işletmen mesajlarını alır. Sonra, bu mesajları SNAServer'a gönderir. SNAServer, aldığı VTAM işletmen mesajlarını ilgili kullanıcı sürecine iletir. Kullanıcı sürecinde çalışmakta olan Kullanıcı Arayüzü, VTAM mesajlarını işleyerek izleme ekranını yeniler veya kullanıcı NCCF ortamında bulunuyorsa, bu mesajları aynen ekrana yansıtır.

Bu kitaptaki Bölüm 2'nin amacı, SNA temel kavramlarını okuyucuya vermektir. Aynı bölümde, VTAM'ın etki-alanı ve ağ kaynaklarının denetiminde kullanılabilen VTAM işletmen komutlarından söz edilmektedir. Bölüm 3'te, VTAM Uygulaması'nın yapısı ve geliştirilmesi için kullandığım makrolar anlatılmaktadır. Bölüm 4'te, SNAServer ve Kullanıcı Arayüzü uygulamalarının yapısı, bu programlarda kullandığım fonksiyonlar ve ağ kaynaklarının izlenmesi ve denetimi için sunulan olanaklardan söz edilmektedir. Ek bölümünde, geliştirdiğim VTAM Uygulaması ile SNAServer arasındaki veri iletişimi için gerekli olan, LU-LU oturumu ile ilgili bazı süreçlerin senaryoları bulunmaktadır.

BÖLÜM 2

2.1 SNA Katmanları

SNA (Systems Network Architecture), çeşitli IBM ürünleri arasında veri iletişimi sağlayan bir ağ yapısıdır.

SNA, 7 katmandan oluşan bir sıradüzenli bilgisayar ağıdır.

1. Katman - PHYSICAL CONTROL LAYER

Bu katman bitişik iki düğümün fiziksel olarak birbirine bağlanmasını sağlar.

2. Katman - DATA LINK CONTROL LAYER

Bu katman iki bitişik düğüm arası veri alış-verişi gerçekleştirir. Bu katmanda SDLC (Synchronous Data Link Control) ve S/370 protokolleri kullanılır.

3. Katman - PATH CONTROL LAYER

Bu katman, veriyi kaynak noktasından, varış noktasına yönlendirir ve ağdaki veri trafiğini kontrol eder.

4. Katman - TRANSMISSION CONTROL LAYER

Bu katman, veri alış-verişinin hız denetimini ('pacing' işlemini) sağlar. Ayrıca, verinin varış noktasına, doğru sırada ulaşıp ulaşmadığını kontrol eder ve güvenlik gerekli ise şifrelemeyi gerçekleştirir.

5. Katman - DATA FLOW CONTROL LAYER

Bu katmanda, veri akışının zaman-uyumluluğu, veri alış-verişinin ilişkilendirilmesi ve ilgili verilerin birimlere gruplandırılması sağlanır.

6. Katman - PRESENTATION SERVICES LAYER

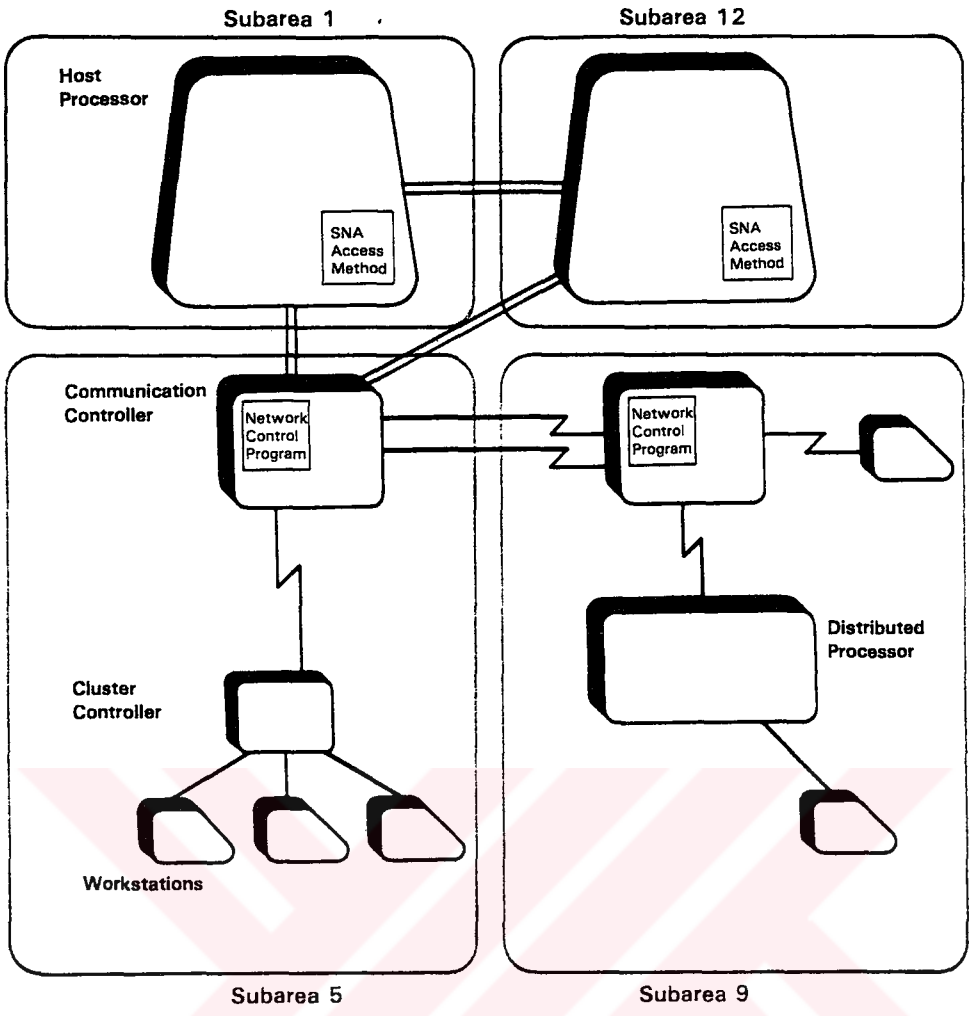
Bu katman, veriyi çeşitli ortamlara uygun şekilde biçimlendirir ve kaynakların paylaşımını koordine eder.

7. katman - TRANSACTION SERVICES LAYER

Bu katman, dağınık veri tabanlarına erişim ve döküman takası gibi uygulama hizmetleri sunar.

2.2 SNA Düğüm Çeşitleri

Bir bilgisayar ağı çok çeşitli donanım ve yazılım bileşenlerinden oluşur. SNA'de bir düğüm, 7 yapısal katmanların fonksiyonlarını gerçekleştiren, donanım ve yazılım bileşenlerin tümüdür.



Şekil 2.1 SNA'de Alt-alan

SNA'de üç çeşit düğüm bulunmaktadır:

1. Anasistem Alt-alan Düğümü (Host Subarea Node)

Anasistem Alt-alan Düğümü, bir SNA erişim yöntemi programı bulunduran düğümdür. Bu düğüm, SNA ağının kontrolü ve yönetimi için gerekli fonksiyonları temin eder.

2. İletişim Denetleyici Alt-alan Düğümü (Communication Controller Subarea Node)

İletişim Denetleyici Alt-alan Düğümü, NCP (Network Control Program) gibi bir ağ denetim programını içeren iletişim denetleyicisidir. Bu çeşit düğümler, veri akışının denetimini ve yöneltilmesini gerçekleştiren fonksiyonları temin eder.

3. Çevresel Düğümler (Peripheral Node)

Çevresel Düğümler, alt-alan düğümü olmayan diğer düğümlerdir. Bunlar, aygıt grubu denetleyicisi veya dağınık işlemci olabilirler.

Bir **alt-alan**, bir alt-alan düğümü ve bu düğüme bağlı çevresel düğümlerden oluşur. Şekil 2.1'de dört adet alt-alan içeren bir SNA ağ örneği bulumaktadır.

Uç-kullanıcılar, bilgisayar ağında gönderilen verinin esas kaynak ve varış noktalarıdır. Uç-kullanıcılar, bir iş-istasyonun aracılığı ile bilgisayar ağına erişen bireyler veya uygulama programları olabilirler.

2.3 NAU - Ağ adreslenebilir Birimler

NAU'lar (Network Addressable Unit- Ağ Adreslenebilir Birim), bir bilgisayar ağındaki uç-kullanıcıların veri alış-verişinde bulunmasına imkan verirler. Her NAU' unun bir adresi vardır. SNA'de üç çeşit NAU tanımlıdır: LU, PU ve SSCP.

1. LU (Logical Unit)- Kavramsal Birim

Her uç-kullanıcı , SNA ağına bir LU 'nun aracılığı ile erişebilir. LU'lar, uç-kullanıcılar arası veri alış-verişini mümkün kılar. Bir LU, bir iş istasyonunu, bir yazıcıyı veya bir uygulama programını temsil edebilir.

2. PU (Physical Unit) - Fiziksel Birim

Her düğümde bir PU bulunur. PU, bulunduğu düğümü bitişik düğümlere bağlayan bağlantıları denetler. Bir PU, bir anasistemi veya bir denetleyiciyi temsil edebilir. PU 'lar , diğer NAU'lar gibi, düğümde bulunan donanım ve yazılım bileşenleri ile gerçekleşir. Bir PU, bir düğümün kaynaklarını ve ona bağlı çevresel aygıtları denetler.

3. SSCP (System Services Control Point) - Sistem Servisleri Denetim Noktası

SSCP, bilgisayar ağındaki kaynakları kontrol eder, oturum kurma sürecini denetler, işletmen komutlarını alır ve işletmen mesajlarını gönderir. SSCP sadece anasistem düğümlerinde bulunur. SSCP, SNA erişim yöntemi programında gerçekleştirilir. En yaygın SNA erişim yöntemi programı, VTAM'dır.

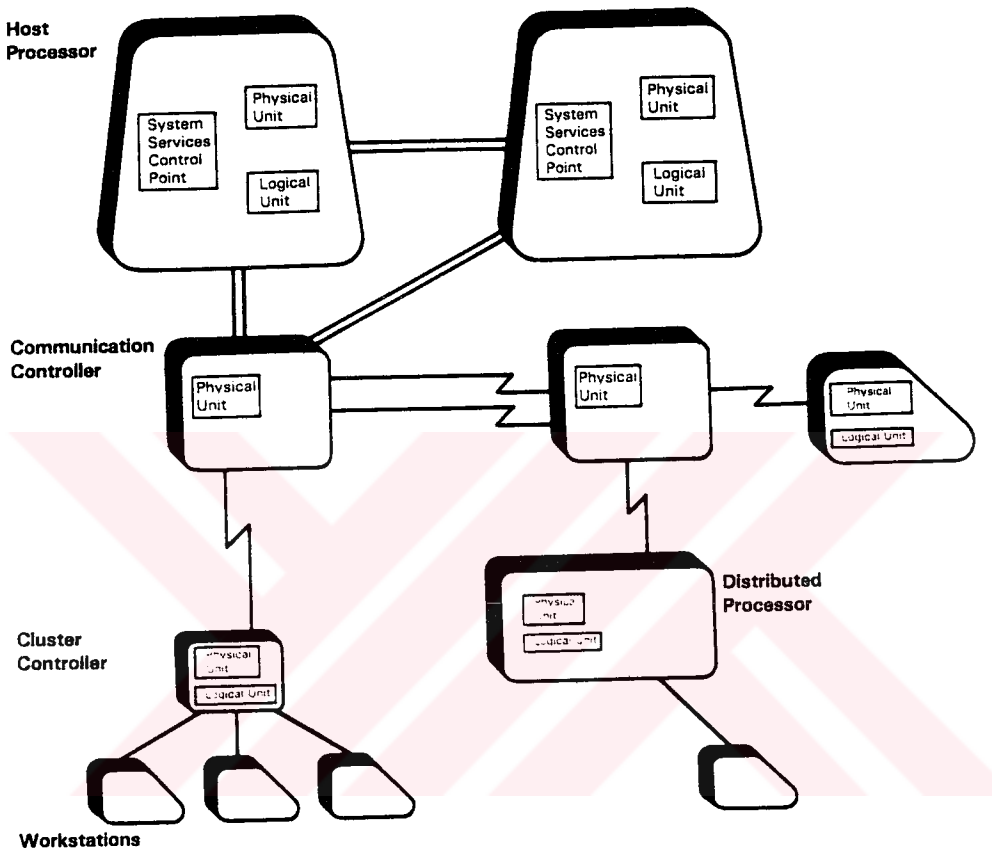
SNA'de üç çeşit PU bulunmaktadır:

- PU tip 5 - anasistem düğümünde bulunur.
- PU tip 4 - iletişim denetleyici düğümünde bulunur ve NCP ağ denetim programı tarafından kontrol edilir. Örnek: IBM 3725, IBM 3745 İletişim Denetleyicileri.
- PU tip 2 - Çevresel Düğümlerde bulunur ve maximum 256 adet LU destekleyebilir.

Şekil 2.2'de değişik düğümlerde bulunabilen NAU'lar gösterilmektedir.

Bir **etki-alanı** (domain), bir SSCP'nin yönettiği ağ bölümüdür. SSCP, kendi etki-alanındaki tüm ağ kaynaklarını denetler. SNA'de iki çeşit ağ bulunur:

1. Tek etki-alanı içeren ağ (Single-domain network) - tek bir SSCP içeren bilgisayar ağıdır.
2. Çok etki-alanı içeren ağ (Multiple-domain network) - birden fazla SSCP içeren bilgisayar ağıdır.



Şekil 2.2 : NAU - Ağ Adreslenebilir Birimler

2.4 NAU Oturumları

Bir oturum (session) - iki NAU arası iletişimi sağlamak için kurulan kavramsal bağlantıdır. SNA'de dört çeşit oturum vardır. Bunlar, SSCP - SSCP, SSCP - PU, SSCP - LU ve LU-LU oturumlarıdır.

1.- SSCP - SSCP oturumları

Bu oturumlar, farklı etki-alanlarda bulunan LU'lar arasındaki oturumların kurulmasını mümkün kılar ve bu LU'lar arası iletişimi denetler.

2.- SSCP - PU oturumları

Bir SSCP-PU oturumu, bir düğümü ve ona bağlı kaynakları denetler.

3.- SSCP - LU oturumları

Bu oturumlar, LU - LU oturumların gerçekleştirilmesi için SSCP'nin arabuluculuk yapmasını sağlar.

4.- LU - LU oturumları

Bir LU - LU oturumu, iki uç-kullanıcı arasında veri iletişimini sağlar. Örnek : CICS uygulaması ve bir iş istasyonu arası iletişim.

Bir SSCP'nin etki-alanında oturumların kurulması farklı aşamalarda gerçekleşir. Önce SSCP, konfigürasyondaki her PU ile bir oturum kurar. Daha sonra SSCP, bir PU altında bulunan her LU için, bir SSCP - LU oturumu kurar. Bir çift LU veri alış-verişinde bulunmak istediğini belirtiğinde, SSCP bu LU'lar arasında bir LU - LU oturumun kurulmasını sağlar.

İki NAU arasındaki iletişim İstek/Yanıt Birimleri (RU - Request/Response Unit) ile yapılmaktadır. Bir İstek Birimi uç-kullanıcı verisi, kontrol bilgisi veya hem veri, hem kontrol bilgisini içeren mesaj birimidir. Yanıt Birimi ise, bir İstek Birimi'nin alındığını veya İstek Birimi'nin başarılı bir şekilde işlenip işlenmediğini bildiren mesaj birimidir. Bu kitapta, bir İstek Birimi'ine kısaca RQ, Yanıt Birimi'ine ise RSP denilmektedir.

2.5 LU - LU Oturumları

İki LU arasında çeşitli oturumlar kurulabilir. Kurulacak oturumun tipi , bu LU'ların özelliklerine bağlıdır. LU-LU oturum tipleri, oturum ortakların tipini ve aralarındaki oturumda kullanılacak protokolleri belirlerler. Bir LU-LU oturumunda, LU'lardan birisi PLU yani Birincil Kavramsal Birim, diğeri ise SLU yani İkincil Kavramsal Birim rolü oynamaktadır. PLU veya SLU rolleri, genellikle önceden tanımlıdır. Örneğin, uçbirim LU'ları sadece SLU olabilmektedir. Uygulama LU'ları ise PLU veya SLU olabilirler.

LU-LU oturum tipleri:

1. Tip 0 oturumları

Bu oturumlarda önceden belirlenmiş protokol tanıtımları (protocol profile) yoktur. Tip 0 oturumlarında, SNA 'in tanımladığı her veri formatı ve protokolleri kullanılabilir veya o uygulamaya has veri formatı ve protokolleri geliştirilebilir. Tip 6 oturumları geliştirilmeden önce bu tip oturumlar iki program arasındaki iletişim için kullanılırdı.

2. Tip 1 oturumları

Tip 1 oturumları, SCS (SNA Character String) veri formatını destekler. Bu oturumların en yaygını, SCS modunda çalışan yazıcıları destekleyen oturumlardır.

3. Tip 2 oturumları

Bu oturum tipi, bir uygulama programı ve 3270 veri formatını kullanan bir görüntüleme aygıtı arasındaki iletişimi destekler.

4. Tip 3 oturumları

Tip 3 oturumu, bir uygulama programı ve 3270 veri formatını kullanan bir yazıcı arasındaki iletişimi destekler.

5. Tip 4 oturumları

Tip 4 oturumu, SCS veri formatını kullanan kelime-işlem uçbirimleri için kullanılan oturum tipidir.

6. Tip 6 oturumları

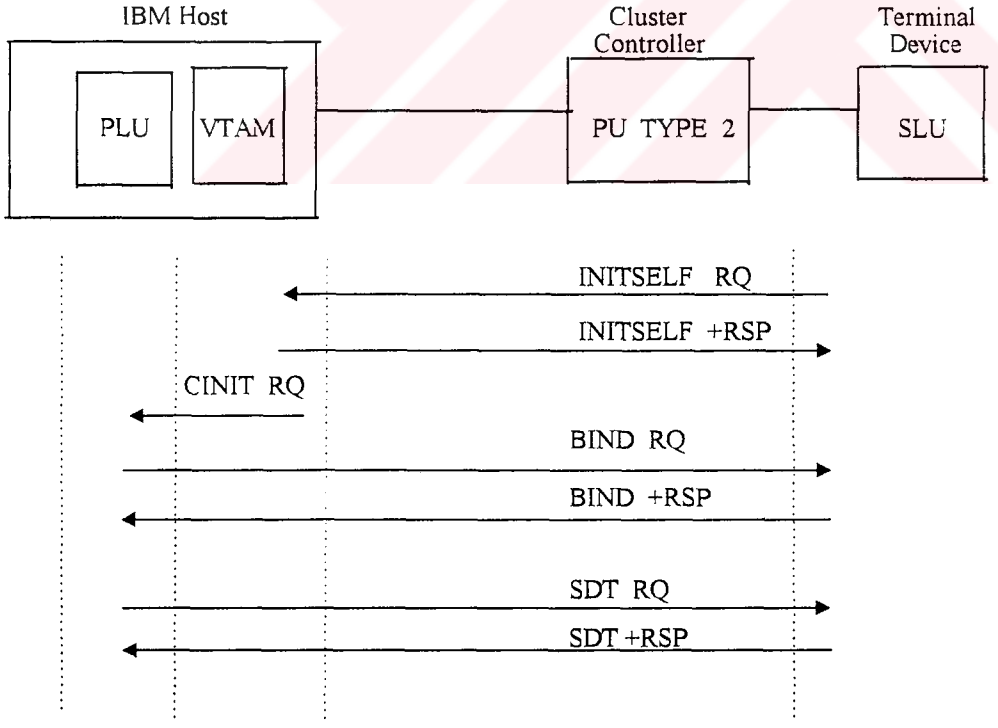
Bu oturumlar, iki hareket programı (transaction) arasındaki iletişimde kullanılan oturumlardır.

7. Tip 7 oturumları

Tip 7 oturumları, 5250 veri formatını kullanan ofis uçbirimleri destekler.

2.6 Bir LU-LU Oturumun Kurulması

Şekil 2.3'te, bir uçbirim aygıtını temsil eden bir SLU ile bir uygulama LU'su (PLU) arasında oturumun nasıl kurulduğu gösterilmektedir.



Şekil 2.3: LU-LU oturumun kurulması

Bu işlem üç aşamada gerçekleşmektedir:

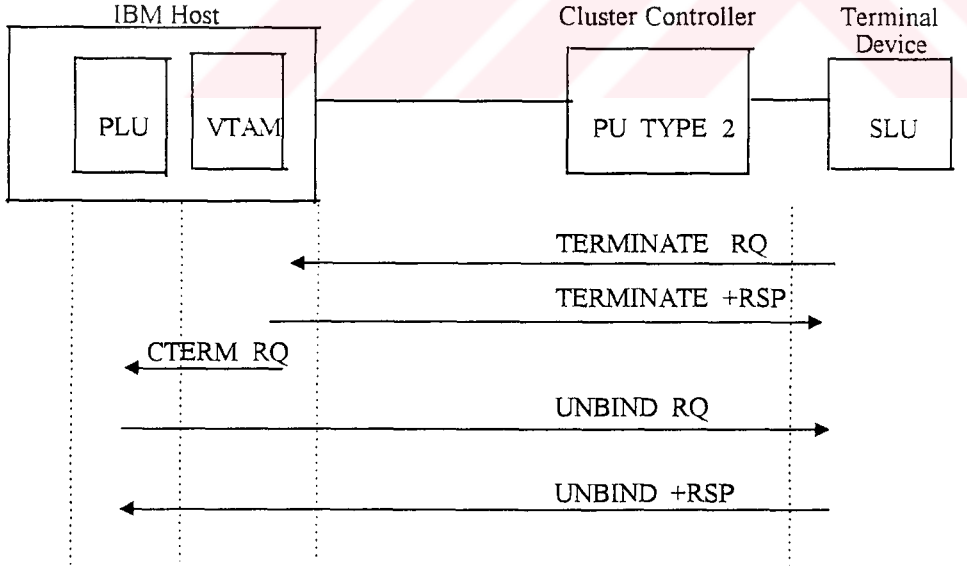
1. Birinci aşamada, SLU, SSCP'ye bir INITSELF RQ'u gönderir. Bu İstek Birimi ile SLU, bir LU-LU oturumun kurulmasını talep eder. SSCP, bu İstek Birimi'ne karşılık olarak bir INITSELF (+RSP) Olumlu Yanıt Birimi'ni SLU'ya gönderir. SSCP, her iki LU'nun, bir oturum kurmak için hazır olup olmadıklarını kontrol eder. Eğer her iki LU hazır ise, SSCP bu iki LU arasında bir **pending-active-session** (etkinleştirilmesi için askıda bekleyen oturum) yaratır.

2. İkinci aşama, **pending-active-session'ın** kurulması ile başlar. SSCP, PLU'ya bir CINIT (Control Initiate) RQ'u gönderir. CINIT RQ, PLU'ya gönderilen, bir oturum kurma talebidir.

3. Üçüncü aşama, PLU'nun SLU'ya bir BIND RQ'ni göndermesi ile başlar. BIND, oturum protokol bilgisini taşır. SLU, BIND RQ'deki oturum parametrelerini inceler. Parametreler uygun ise SLU, PLU'ya bir BIND (+RSP) Olumlu Yanıt Birimi'ni gönderir. PLU, bu Yanıt Birimi'ni aldığında oturum kurulmuş olur. Eğer SLU, BIND RQ'daki parametreleri uygun bulmazsa bir Olumsuz Yanıt Birimi'ni (-RSP'i) gönderir ve LU - LU oturumu kurulamamış olur.

4. LU-LU oturumu kurulduktan sonra, iki LU arasında veri alış-verişin başlayabilmesi için, PLU bir SDT (Start Data Traffic) İstek Birimini gönderir ve buna karşılık bir Olumlu Yanıt Birimi'ni alır.

2.7 Bir LU-LU Oturumun Sonlandırılması



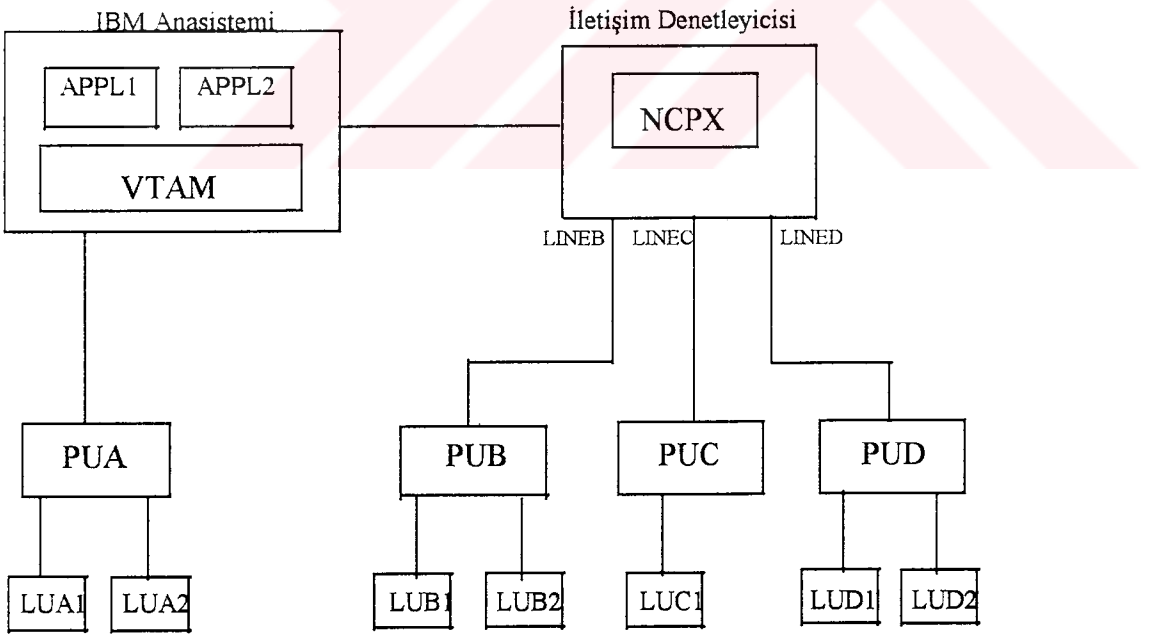
Şekil 2.4: LU-LU oturumun sonlandırılması

Şekil 2.4, bir uçbirim aygıtını temsil eden bir SLU ve bir uygulama LU'su (PLU) arasındaki oturumunun nasıl sonlandırıldığını göstermektedir.

1. SLU, bir Terminate İstek Birimi'ni SSCP'ye göndererek, LU-LU oturumun sonlandırılması için istekte bulunur.
2. SSCP, bu İstek Birimi'ne karşılık olarak SLU'ya bir Olumlu Terminate Yanıt Birimi'ni gönderir.
3. SSCP, SLU'nun isteğini PLU'ya bir CTERM (Control Terminate) İstek Birimi ile bildirir.
4. PLU, CTERM İstek Birimi'ni aldıktan sonra, SLU'ya bir UNBIND İstek Birimi'ni gönderir.
5. PLU, bu İstek Birimi'ne karşılık bir Olumlu Yanıt Birimi (UNBIND +RSP) aldığında LU-LU oturumu sona erer.

2.8 VTAM Etki-alanı

VTAM, SNA ortamında gerekli olan erişim yöntemi fonksiyonlarını sağlar. VTAM, anasistem düğümünde çalışan bir programdır ve SSCP'yi içerir. VTAM, iletişim denetleyicilerinde bulunan NCP programı ile birlikte SNA ağının denetimini sağlar.



Şekil 2.5: Bir SNA etki-alanı

VTAM, etki-alanında bulunan kaynakları kontrol eder. Şekil 2.5'de bir SNA etki-alanı gösterilmektedir. Bu ağ örneğinde, VTAM'ın kontrol ettiği kaynaklar arasında bir NCP Büyük Düğüm'ü (Major Node'u), tip 2 PU'lar, çevresel LU'lar ve uygulama LU'ları (APPL1 ve APPL2) bulunmaktadır.

VTAM'da, ortak özellikler taşıyan ağ kaynakları **kavramsal** olarak Büyük Düğüm'lere (Major Node'lara) gruplandırılmaktadır. Bir Büyük Düğüm, bir veya birden fazla kaynak kapsamaktadır. Bu kaynaklara Küçük Düğüm (Minor Node) denmektedir.

Temel Büyük Düğüm tipleri:

1. Yerel Büyük Düğümü

Bir Yerel Büyük Düğümü, IBM anasistemine direkt bağlı olan çevresel düğümleri içermektedır. Örneğin, Şekil 2.5'de görülen PUA ismindeki PU'yu ve altındaki LU'ları (LUA1 ve LUA2) içeren bir Yerel Büyük Düğümü tanımlanabilir.

2. Uygulama Büyük Düğümü

Bir Uygulama Büyük Düğümü, bir grup VTAM uygulamasını içermektedir. Örneğin, Şekil 2.5'de görülen APPL1 ve APPL2 uygulamalarını içeren bir Uygulama Büyük Düğümü tanımlanabilir.

3. NCP Büyük Düğümü

Bir NCP Büyük Düğümü, bir iletişim denetleyicisini ve ona bağlı uzak ağ kaynaklarını icerir. Şekil 2.5` da görülen NCPX , bir NCP Büyük Düğümü örneğidir. NCPX, üç adet hat (LINEB, LINEC ve LINED), bu hatların uçlarında bulunan PU'ları (PUB, PUC ve PUD) ve PU'lar altındaki LU'ları içermektedir.

2.9 VTAM İşletmen Komutları

Ağ kaynaklarını izlemek ve denetlemek için VTAM işletmen komutları kullanılmaktadır. Bir VTAM komutu girildiğinde, karşılığında VTAM işletmen mesajları alınır. VTAM mesajları, bir komutun kabul edildiğini ve komutta belirtilen işlemin nasıl sonuçlandığını bildirir. Birçok VTAM mesajı soruşturulan kaynağın durumunu içermektedir. Örnek durum kodları, aşağıda verilmektedir.

Kaynak durum kodları :

ACTIV - resource active - kaynak etkin durumunda

INACT - resource inactive - kaynak işlem-dışı

IINOP - resource inoperative - kaynak işlem-dışı

CONCT - resource connectable - kaynak bağlanabilir

PCTD1 - resource pending 1 - kaynak askıda, etkinleştirilmek üzere

PCTD2 - resource pending 2 - kaynak askıda, etkinleştirilmek üzere

Dört çeşit VTAM komutu bulunmaktadır: DISPLAY, VARY, MODIFY ve HALT.

1. DISPLAY (D)

DISPLAY komutları ağ kaynakları hakkında ayrıntılı bilgi edinmek için kullanılmaktadır.

Örnekler:

D NET,ID=LINEB,E

- komutu LINEB kaynağı ve altında bulunan kaynaklar hakkında ayrıntılı bilgi verir.

D NET,LINES

- komutu etki-alanında bulunan tüm hatlar hakkında bilgi verir.

D NET,MAJNODES

- komutu etki-alanında bulunan tüm Büyük Düğüm'ler hakkında bilgi verir.

D NET,APPLS

- komutu etki-alanında bulunan tüm uygulamalar hakkında bilgi verir.

D NET,CLSTRS

- komutu etki-alanında bulunan tüm aygıt grubu denetleyicileri (PU tip 2) hakkında bilgi verir.

2. VARY (V) komutları

Bu komutlar, ağ kaynaklarının kontrolü için kullanılırlar.

VARY komutların bazı çeşitleri : ACT, INACT, LOGON, NOLOGON ve TERM dır.

2.a. ACT- Bu komut, kaynakları etkin duruma getirme komutudur.

Örnek:

V NET,ACT,ID=LUB1,LOGON=CICSA

Yukarıdaki komut, LUB1 kavramsal biriminin etkin durumuna getirilmesini ve CICSA uygulaması ile LU-LU oturumun kurulmasını sağlar. Başka bir değişle, bu komutla, LUB1'in, CICSA'ya 'logon' olması istenmektedir.

2.b. INACT -Bu komut, kaynakları işlem-dışı durumuna getirir.

Örnek:

V NET,INACT,ID=LUB1,F

2.c. LOGON - Bu komut, bir LU için otomatik 'logon' tanımını yaratır veya bu tanımı değiştirir.

Örnek:

V NET,LOGON=CICSAPPL,ID=LUB1

Yukardaki komut, LUB1 kavramsal biriminin, CICSAPPL uygulamasına 'logon' olmasını sağlamaktadır.

2.d. NOLOGON - Bu komut, bir LU'nun otomatik 'logon' tanımını siler.

Örnek:

V NET,NOLOGON,ID=LUB1

2.e. TERM - Bu komut, bir oturumu veya bir oturum grubunu sonlandırır.

Örnek:

V NET,TERM,LU1=CICSA,LU2=LUB1

Bu komut, CICSA uygulaması ve LUB1 kavramsal birimi arasındaki LU-LU oturumunu sonlandırır.

3. MODIFY (F) VTAM

Bu komutlar, VTAM parametrelerinin dinamik olarak değiştirilmesini sağlarlar.

Örnekler:

F NET,TRACE,TYPE=LINE,ID=LINEC

Yukardaki komut, LINEC ismindeki kaynak için 'trace' işlemini başlatır.

F NET,NOTRACE,TYPE=LINE,ID=LINEC

Yukardaki komut, bir 'trace' işlemini durdurur.

F NET,TABLE,ID=TABXX,OPTION=DELETE,TYPE=MODETAB

Yukardaki komut ise, bir 'logmode' tablosunu siler.

4. HALT (Z)

HALT komutu VTAM erişim yöntemi sistemini durdurmak için kullanılır.

BÖLÜM 3

3.1 VTAM Programlama

Bir uygulama programı, VTAM`ın hizmetlerinden yararlanabilmesi için VTAM makrolarını kullanmalıdır.

VTAM makroları aşağıdaki işlemleri sağlamaktadır:
- Uygulama programı ve VTAM arasında bir ilişkinin kurulmasını veya bu ilişkinin koparılmasını,
- Uygulama programı ve belirli LU'lar arasında oturumların kurulmasını ve sonlandırılmasını ,
- LU'larla veri alış-verişini,
- VTAM hizmetlerini talep ederken kullanılan kontrol blokların tanımlanmasını ve ilk değerlerin atanmasını,
- Bir kontrol bloğunun işlenmesini,
- VTAM işletmen komutlarının ve mesajlarının, uygulama programı ve VTAM arasında transferini sağlamaktadırlar.

VTAM makroları üç çeşittir:
- bildirim makrolar (declarative macroinstructions)
- ACB-tabanlı makrolar (ACB-based macroinstructions)
- RPL-tabanlı makrolar (RPL-based macroinstructions)

3.2. Bildirim Makroları

ACB, EXLST, NIB ve RPL bloklarının tanımını ve ilk değer atanmalarını gerçekleştiren makrolardır.

1. ACB makrosu

Bu makro bir ACB (access method control block) bloğunu tanımlar ve ilk değerlerini belirler. Bir ACB bloğu, VTAM'a uygulama programı hakkında bilgi verir. Temel olarak bir ACB bloğu, uygulamayı adlandırır ve onunla ilgili EXRT'lerin (exit routine - çıkış program parçası) listesini bildirir.

2. EXLST makrosu

Bu makro, bir EXLST (exit list) bloğunu tanımlar. Bir EXLST bloğu, uygulama programındaki EXRT'lerinin adreslerini içerir. Uygulama programı, EXRT'lerin aracılığı ile belli olayları öğrenip kontrol edebilir. Belirli olaylar meydana geldiğinde, VTAM bu EXRT'leri çalıştırır.

3. NIB makrosu

Bu makro bir NIB (node initialization block) bloğunu tanımlar ve ilk değerlerini belirler. Bir NIB bloğu, bir LU-LU oturumu hakkında protokol bilgisi içerir. Bu bilgiler, bir LU-LU oturumun kurulma aşamasında verilir ve oturum sonlandırılıncaya kadar geçerli olur.

4. RPL makrosu

Bu makro bir RPL (request parameter list) bloğunu tanımlar ve ilk değerlerini belirler. Bir RPL bloğu, bir uygulama programının, RPL-tabanlı işlemlerini talep ederken, VTAM'a ilettiği bilgileri (parametreleri) içerir. Talep edilen işlem tamamlandıktan sonra VTAM, uygulama programı için RPL bloğuna bilgiler yerleştirir. Bir RPL bloğu, bir işlem talebi ve işlemin sonucu hakkında bilgi içerir.

3.3 ACB-tabanlı Makrolar

1. OPEN makrosu

Bu makro, uygulama programını VTAM'a tanıtır ve programın, VTAM hizmetlerini kullanacağını bildirir. Sadece bu tanıtımdan sonra VTAM, bu uygulama programı ile ilgili oturum kurma taleplerini kabul eder ve belli koşullar meydana geldiğinde programdaki EXRT'leri çalıştırır.

2. CLOSE makrosu

Bu makro, uygulama programı ile VTAM arasındaki ilişkinin sonlandırılmasını talep eder. Bu makro işlendikten sonra, uygulama programı VTAM hizmetlerini kullanamaz.

3.4 RPL-tabanlı Makrolar

Bu makroların hepsi bir RPL bloğu kullanır.

RPL-tabanlı makroların çeşitleri:

-1. oturum kurma makroları

-2. oturum sonlandırma makroları

-3. veri alış-veriş makroları

-4. oturum kurma ve veri alış-verişinde yardımcı makrolar

-5. program işletmen makroları

3.4.1. Oturum Kurma Makroları

OPNDST, REQSESS, SIMLOGON, OPNSEC oturum kurma makrolarıdır. Bu tezdeki LU-LU oturumun kurulması için OPNDST makrosunu kullandım. Kurulan oturumda, geliştirdiğim VTAM uygulama programı PLU rolünde olmaktadır.

3.4.2. Oturum Sonlandırma Makroları

CLSDST, SESSIONC ve TERMSESS oturum sonlandırma makrolarıdır. Geliştirdiğim VTAM uygulama programında CLSDST makrosunu kullandım. CLSDST makrosu, bir LU-LU oturumun sonlandırılmasını VTAM'dan talep eder veya bir oturum kurma talebini (CINIT RQ'ini) geri çevirmek için VTAM'a istekte bulunur.

3.4.3. Veri Alış-veriş Makroları

Bu makrolar, bir LU-LU oturumu üzerinden veri alış-verişi gerçekleştiren makrolardır.

1. RECEIVE makrosu

Bu makro, uygulama programına bir LU-LU oturumu üzerinden gelen bir RU'nun (İstek/Yanıt Birimi'nin) veri kısmını programın veri giriş alanına ve kontrol bilgisi içeren kısmını RPL bloğunun belirli alanlarına aktarılması için VTAM'a talepte bulunur.

2. SEND makrosu

Bu makro, bir RU'nun (İstek/Yanıt Birimi'nin) belirtilen LU-LU oturumu üzerinden gönderilmesi için VTAM'a talepte bulunur. RU'nun veri kısmı uygulama programının veri çıkış alanından alınır. Kontrol bilgisini içeren kısmı ise SEND makrosunda parametre olarak belirtilir.

3.4.4. Yardımcı Makrolar

1. CHECK makrosu

Daha önce talep edilen, zaman-uyumsuz bir RPL-tabanlı işleminin tamamlanıp tamamlanmadığını kontrol eder ve gerekli ise tamamlanmasını bekler. RPL-tabanlı işleminde kullanılan RPL bloğunu, yeniden kullanılabilmesi için işaretler.

2.. INQUIRE makrosu

Bu makro, uygulama programının ihtiyacı olan, belirli bilgileri elde edip belirtilen program alanına yerleştirir. Bu bilgiler şunlar olabilir:

- oturum başlatma talebi ile ilgili kullanıcı bilgisi (Örneğin: kullanıcı ismi ve şifresi)
- kurulması için askıda bekleyen bir oturumun, oturum protokollerini belirten parametreler.

3. SETLOGON makrosu

Bu makro aşağıdaki şekillerde kullanılabilir:

* SET LOGON OPTCD=START

Uygulama için CINIT RQ'ları alındığında, uygulamanın LOGON tipi EXRT'nin çalıştırılabileceğini VTAM'a bildirir.

* SETLOGON OPTCD=HOLD

Uygulama programı için CINIT RQ'ları alındığında, LOGON tipi EXRT'nin çalıştırılmaması gerektiğini VTAM'a bildirir.

3.4.5. Program İşletmen Makroları

Bir Program İşletmeni (Program Operator) yetkili bir uygulamadır. Bir Program İşletmeni, VTAM'a VTAM işletmen komutlarını iletir ve VTAM'dan işletmen mesajlarını alır.

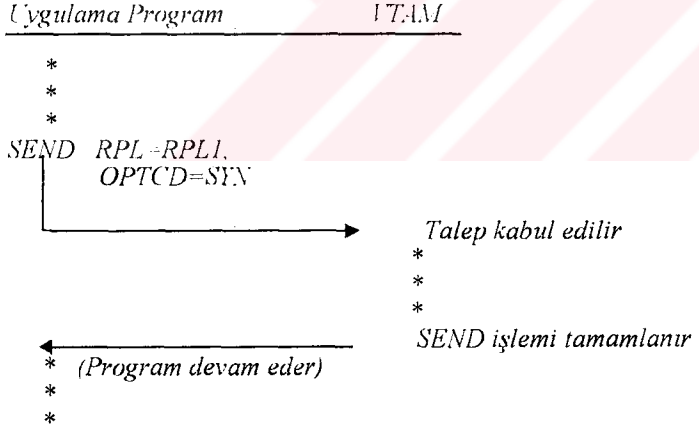
1. SENDCMD makrosu

Bu makro, bir VTAM işletmen komutunun VTAM'a verilmesini sağlar.

2 RCVCMD makrosu

Bu makro, SENDCMD makrosu ile VTAM'a verilen komutların cevaplarını, yani VTAM mesajlarını almak için kullanılır.

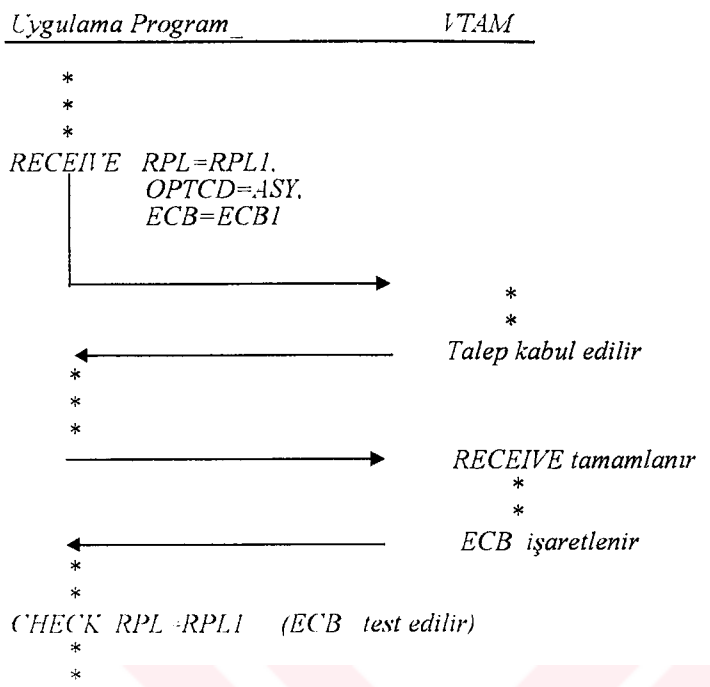
3.5. Zaman-uyumlu/Zaman-uyumsuz İşlemler



Şekil 3.1 : Zaman-uyumlu işlem örneği

Bir VTAM uygulama programı, VTAM'dan bir işlem talep ederken, bu işlemin zaman-uyumlu veya zaman-uyumsuz olarak gerçekleşmesini isteyebilir. İstenilen işlem türü, [OPTCD=SYN/ASY] RPL parametresi ile belirtilir.

Zaman-uyumlu bir işlem talep edildiğinde, VTAM, bu işlem tamamlanmadan, kontrolü programın bir sonraki komutuna devretmez. Şekil 3.1'de, zaman-uyumlu bir SEND işlemi gösterilmektedir.



Şekil 3.2: Zaman-uyumsuz işlem örneği

Bir zaman-uyumsuz işlem talep edildiğinde, VTAM, talep edilen işlem tamamlanmadan talebi kabul eder etmez kontrolü, programın bir sonraki komutuna devreder. Talep edilen işlem devam ederken, uygulama programı başka işlemler yapabilmektedir. Zaman-uyumsuz bir işlem talep edildiğinde, bu işlemin tamamlandığını programa bildirmek için iki farklı yöntem kullanılabilir.

Birinci yöntemde, uygulama programı talep ettiği zaman-uyumsuz işlemi bir ECB (event control block) bloğu ile ilişkilendirir. VTAM bu işlem tamamlandığı zaman, ECB bloğunu işaretler. Bu şekilde, ECB bloğuna işlemin tamamlandığını gösteren bir değer atanır. Daha sonra program bir CHECK makrosu ile ECB bloğunu kontrol eder. Bu yöntem Şekil 3.2 'de gösterilmektedir.
İkinci yöntemde, uygulama programı VTAM'a bir zaman-uyumsuz işlemi talep ederken, bu işlemi bir RPL tipi EXRT ile ilişkilendirir. İşlem tamamlandığında ise VTAM, belirtilen RPL EXRT'sini çalıştırır.
Bir zaman-uyumsuz işlemini, bir ECB bloğu veya bir RPL EXRT'si ile ilişkilendirmek için RPL bloğundaki, [ECB/EXIT] parametresine uygun değer atanmalıdır.

Bu tezde geliştirdiğim VTAM uygulama programında, çoğu VTAM makrosu, zaman-uyumlu işlemler talep etmektedir. SNAServer programı ile veri iletişimi gerçekleştiren VTAM makroları ise, zaman-uyumsuz işlemler talep etmektedirler.

3.6. EXLST Çıkış Program Parçaları

EXLST EXRT'leri, özel amaçlı program parçalarıdır. Uygulama programı için önemli olan, belirli olaylar meydana geldiğinde VTAM, programın EXRT'lerini çalıştırır. Bir uygulamada bulunabilen EXLST tipi EXRT'ler şunlardır: DFASY, LERAD, LOGON, LOSTERM, NSEXIT, RELREQ, RESP, SCIP, SYNAD ve TPEND. Bu tezdeki VTAM uygulama programında, sadece LOGON, RESP, TPEND ve LOSTERM tipi EXRT'lere gerek duydum..

1. LOGON tipi EXRT

VTAM, bir CINIT RQ'nun geldiğini programa bildirmek için, programın LOGON tipi EXRT'sini çalıştırır.

2. LOSTERM tipi EXRT

VTAM, programın LOSTERM tipi EXRT'sini, bir LU-LU oturumunun sonlandırıldığını veya sonlandırılması gerektiğini, programa bildirmek için çalıştırır.

3. TPEND tipi EXRT

VTAM, programın TPEND tipi EXRT'sini aşağıdaki durumlarda çalıştırır.

- VTAM hatalı bittiğinde (abend),
- VTAM, iletişim işletmeni tarafından kapatılırken (HALT komutu ile),
- iletişim işletmeni VTAM uygulamasını kapatırken (işlem-dışı durumuna getirirken).

4. RESP tipi EXRT

Uygulama programına, LU-LU oturumu üzerinden bir Yanıt Birimi geldiğinde VTAM, programın RESP tipi EXRT'sini çalıştırır.

3.7. VTAM APPL Tanımları

Bu tezde, VTAM uygulama programı için iki APPL (uygulama) LU'su tanımlandım. Bu APPL LU'ların ismi VTAPL ve VTSPO dur. Bu LU'ların, SNA ağında yer alabilmeleri için, önce aşağıda görülen VTAM tanımlarını yaptım.

```
VTAPL   APPL   AUTH=(PASS,ACQ)          (3.7.1)
VTSPO   APPL   AUTH=(PASS,ACQ,SPO)      (3.7.2)
```

VTSPO, bir SPO'dur yani İkincil Program İşletmeni'dir. VTSPO, VTAM'a işletmen komutlarının verilmesi ve VTAM'dan işletmen mesajların alınması için kullanılmaktadır. VTAPL ise, SNAServer Programı ile bir LU-LU oturumu uzerinden veri alış-verişinde bulunmaktadır. Bu oturumda VTAPL,

PLU (Birincil Kavramsal Birim) rolünü üstlenirken, SNAServer programı temsil eden LU, SLU (İkincil Kavramsal Birim) rölündedir.

3.8. Kontrol Blok Tanımları

VTAM uygulama programında, VTAPL ve VTSPO için birer ACB ve EXLST blokların tanımlanması için, aşağıdaki deyimlerde görülen bildirim makrolarını kullandım.

Bu makrolardaki bazı parametreler:

* [AM] parametresi, erişim yönteminin VTAM olduğunu ifade etmektedir.

* ACB makrosundaki [APPLID] parametresi, uygulama LU'sunun ismini vermektedir.

* ACB makrosundaki [EXLST] parametresi, uygulamanın Çıkış Liste'sinin (Exit List'esinin) ismini vermektedir.

(3.8.3)'deki bildirim makrosu, VTAPL'nin dört adet EXRT'si olduğunu göstermektedir. Bunlar LOGON, LOSTERM, TPEND ve RESP tipi EXRT'leridir. (3.8.6)'daki bildirim makrosu ise, VTSPO'nun bir tek EXRT'si olduğunu göstermektedir. VTAPL ve VTSPO uygulama LU'larının TPEND tipi EXRT'leri ortaktır.

```
PACB     ACB     AM=VTAM,                    (3.8.1)
                 APPLID=NAME,
                 EXLST=EXLST1,
                 MACRF=LOGON
NAME     DC      AL1(L'NAMEEF)               (3.8.2)
NAMEEF   DC      C'VTAPL'
EXLST1   EXLST   AM=VTAM,                    (3.8.3)
                 LOGON=LOGON1,
                 LOSTERM=LOST1,
                 TPEND=TPEND1,
                 RESP=RESP1

ACBSPO   ACB     AM=VTAM,                    (3.8.4)
                 APPLID=SPONAM,
                 EXLST=EXTSPO
SPONAM   DC      AL1(L'SPONAME)              (3.8.5)
SPONAME  DC      C'VTSPO'
EXTSPO   EXLST   AM=VTAM,                    (3.8.6)
                 TPEND=TPEND1

         *
         OPEN    ACB                         (3.8.7)
         OPEN    ACBSPO                      (3.8.8)
         *
         *
         CLOSE   PACB                        (3.8.9)
         CLOSE   ACBSPO                      (3.8.10)
```

VTAPL ve VTSPO ile ilgili RPL-tabanlı işlemlerinin yapılabilmesi için birer RPL bloğu tanımlanmalıdır. Bu sebeple, VTAPL için (3.8.11), VTSPO için ise (3.8.12) bildirim makrolarını kullandım.

```
RPL1     RPL    AM=VTAM,          (3.8.11)
                ACB=PACB
RPLSPO   RPL    AM=VTAM,          (3.8.12)
                ACB=ACBSPO
```

3.9 OPEN ve CLOSE İşlemleri

OPEN makrosu, VTAM'ın sunduğu olanaklardan yararlanabilmesi için, uygulama programını VTAM'la ilişkilendirir. Program, OPEN makrosu ile bir ACB bloğunu açar. Bir program, birden fazla ACB bloğunu açabildiğinden, VTAM tarafından iki veya ikiden fazla APPL (uygulama LU'su) olarak görülebilmektedir. Bu tezin VTAM uygulama programı VTSPO ve VTAPL isimlerini taşıyan iki APPL olarak görülmektedir.

Bir ACB bloğunun OPEN edilmesi, program ile APPL'nin VTAM tanımı (3.7.1 veya 3.7.2) arasında ilişkinin kurulması demektir. OPEN makrosu, SSCP ve APPL LU'su arasında bir SSCP-LU oturumun kurulmasını sağlamaktadır. Ayrıca, EXLST bildirim makrosu ile tanımlanan EXRT'ler, VTAM tarafından çalıştırılacak durumuna gelmektedirler.

CLOSE makrosu bir APPL'in ACB bloğunu kapatma makrosudur. Bu makro, var olan tüm LU-LU oturumlarını sona erdirir; ayrıca APPL ile VTAM arasındaki ilişkiyi koparır. Bu makronun işlemi tamamlandıktan sonra program, VTAM'ın sunduğu olanaklardan yararlanamaz, çünkü APPL işlem-dışı durumuna getirilmiştir.

Bu tezdeki VTAM uygulama programı, ACB bloklarını kapatması gerektiğini üç yoldan öğrenebilir:

1. Sistem işletmeni, konsoldan programı adresleyen, bir SHUTDOWN mesajı girince
2. Kullanıcı Arayüz Programını kullanan iletişim sorumlusu, LU-LU oturumu üzerinden bir SHUTDOWN mesaji gönderince
3. Programın TPEND tipi EXRT'si VTAM tarafından çalıştırılınca.

Geliştirdiğim VTAM uygulama programının, konsoldan girilen bir mesajla sona erdirilebilmesi icin, program içinde OCIR tipi bir EXRT (Operator Communication Interruption Routine) yazdım. Konsoldan, uygulama programını adresleyen bir MSG komutu girildiğinde, işletim sistemi kontrolü OCIR'e verir. OCIR, konsole mesajını kontrol eder. Bu mesaj bir SHUTDOWN komutu ise, ECBCLOSE isminde bir ECB bloğunu işaretler ve OCIR'den çıkılır. Uygulama programının ana

bölümünde, ECBCLOSE bloğu belli aralıklarla kontrol edilmektedir. Program, bu ECB bloğunun işaretlendiğini fark edince ACB bloklarını CLOSE eder ve sona erer.

VTAM, geliştirdiğim programın TPEND tipi EXRT'sini aşağıdaki nedenlerden dolayı çalıştırabilir :

1. Uygulama programının APPL LU'ları, işletmen tarafından girilen bir komutla işlem-dışı durumuna getirilmek istenildiğinde
2. VTAM'ı kapatmak amacıyla bir HALT komutu girildiğinde
3. VTAM' da oluşan iç hatalardan dolayı, VTAM kendi kendini kapattığında.

Uygulama programının TPEND tipi EXRT'si, programın ECBCLOSE ismindeki ECB bloğunu işaretler. Program, bu ECBCLOSE bloğunun işaretlendiğini fark edince, ACB bloklarını CLOSE eder ve sona erer.

3.10 APPL'nin Oturum Kurma İşlemi

VTAM uygulama programı ve SNAServer programı arasındaki veri alış-verişi, VTAPL ile SNAServer Programını temsil eden LU arasında kurulan bir LU-LU oturumu üzerinden gerçekleşmektedir. Bu oturumda, VTAPL PLU rölündedir, SNAServer Programı ise SLU rolündedir. VTAPL ile SLU arasındaki oturum, BÖLÜM 1'de anlatılan şekilde kurulmaktadır.

Oturum kurma aşamaları:

1. SLU, SSCP'ye bir INITSELF RQ'ü göndererek, VTAPL ile bir LU-LU oturumun kurulmasını talep eder. SSCP, bu RQ'ye karşılık bir INITSELF (+RSP)i gönderir.
2. SSCP, SLU'nun oturum kurma talebini bildirmek için, VTAPL'ye bir CINIT RQ'su iletir.
3. VTAM, bu CINIT RQ'sunu VTAPL'ye aktarmak amacıyla, uygulama programının LOGON tipi EXRT'sini çalıştırır.
4. Uygulama programının LOGON tipi EXRT'sinde, SLU'nun ismi öğrenilir, NIB (3.10..2) ve RPL (3.10.1) blokları hazırlanır ve OPNDST (3.10.3) makrosu işlenir. OPNDST makrosunda belirtilen NIB bloğu, kurulacak olan LU-LU oturumu için protokol bilgisini içerir.

VTAM, OPNDST makrosunu işlerken, NIB'te belirtilen protokol bilgisini kullanarak bir BIND RQ'ü hazırlar, ve onu VTAPL`nin adına SLU'ya gönderir. SLU, BIND RQ'sünde bulunan protokol bilgisini inceledikten sonra, VTAPL'ye bir BIND (+RSP)'ni gönderir. Bu Yanıt Birimi VTAPL'ye ulaşınca VTAM, VTAPL'nin adına SLU2ya bir SDT RQ'ü gönderir ve buna karşılık SDT (+RSP) Olumlu Yanıt

Birimi'ni alır. OPNDST makrosunun işlemi tamamlanır ve VTAPL ile SLU arasındaki oturum etkin hale gelir. Bu aşamadan sonra, VTAPL ve SLU arasında veri alış-verişi gerçekleşebilir.

```
RPLCONN  RPL     AM=VTAM                          (3.10.1)
NIBCONN  NIB     MODE=RECORD,                     (3.10.2)
                   PROC=(RESPX,TRUNC)
                   NAME=SLUNAME
         OPNDST  ACB=PACB,                        (3.10.3)
                 RPL=RPLCONN,
                 NIB=NIBCONN,
                 OPTCD=(ACCEPT,SYN,SPEC)
```

3.11 APPL'nin Oturum Sonlandırma İşlemi

VTAPL ve SLU arasındaki LU-LU oturumu değişik yollardan sona erdirilebilir:

1. SLU, SSCP'ye bir Terminate RQ'ü göndererek oturumu kapatma talebinde bulununca (VTAM programın LOSTERM tipi EXRT'sini çalıştırır)
2. İletişim hattında bir kopukluk meydana gelince (VTAM programın LOSTERM tipi EXRT'sini çalıştırır)

SLU'nun LU-LU oturumunu sonlandırma talebi üzerine şu işlemler gerçekleşmektedir:

1. SLU, SSCP'ye bir Terminate RQ'ini göndererek, oturumun sonlandırılmasını ister. Buna karşılık, SSCP, SLU'ya bir Terminate (+RSP)'ni gönderir.
2. SSCP, SLU'nun oturum sonlandırma talebini VTAPL'ye bildirmek için, bir CTERM RQ'ini iletir.
3. VTAM ise, CTERM İstek Birimi'ni uygulama programına aktarma amacıyla, programın LOSTERM tipi EXRT'sini çalıştırır.
4. VTAM uygulama programının LOSTERM tipi EXRT'sinde, LU-LU oturumu sonlandırılması için (3.11.1) deki CLSDST makrosunu kullandım.

```
CLSDST   RPL=RPL1,                    (3.11.1)
         OPTCD=SYN
```

3.12 APPL'nin Veri Alış-Veriş İşlemleri

VTAPL ile SLU arasındaki veri alış-verişi RU'larla (İstek/Yanıt Birim'leri ile) gerçekleşmektedir.

VTAM uygulama programında kullandığım, (3.12.1) RECEIVE makrosu, SLU'dan gelen RQ'nun INPAREA alanına yerleştirilmesi için VTAM'a talepte bulunur. Bu RECEIVE işlemi zaman-uyumsuz bir işlemdir. VTAM, işlem tamamlandığında RCVECB bloğunu işaretler. Uygulama programı,

RCVECB bloğunu test edip RECEIVE işlemin tamamlandığını öğrendikten sonra, (3.12.4)'deki SEND makrosu ile bir RSP'nin gönderilmesini talep eder.

```
RECEIVE  RPL=RPL1,                          (3.12.1)
         OPTCD=(ASY,Q,ANY),
         ECB=(RCVECB),
         AREA=INPAREA,
         AREALEN=72
*
*
WAITM    RCVECB,ECBCLOSE                    (3.12.2)
*
CHECK    RPL=RPL1                           (3.12.3)
*
*
SEND     RPL=RPL1,                          (3.12.4)
         STYPE=RESP,
         RESPOND=(NEX,FME,NRRN),
         OPTCD=SYN
*
```

Verini VTAPL'den SLU'ya gönderilmesi için (3.12.5)'deki SEND makrosunu kullandım. Bu makro zaman-uyumlu bir işlemdir ve OUTAREA alanındaki verinin bir RQ birimi olarak SLU'ya gönderilmesini talep etmektedir. VTAM, bu RQ'ya yanıt olarak SLU tarafından gonderilen RSP'yi programa iletmek için programın RESP tipi EXRT'sini çalıştırır.

```
SEND     RPL=RPL1                           (3.12.5).
         STYPE=REQ,
         RESPOND=(NEX,FME,NRRN),
         POST=SCHED,
         OPTCD=SYN,
         AREA=OUTAREA,
         RECLEN=OUTLEN
```

3.13 SPO İşlemleri

VTSPO uygulama LU'su, bir SPO'dur. SPO, yetkili bir uygulamadır. SPO, VTAM'a işletmen komutlarını iletir ve VTAM'dan işletmen mesajlarını alır.

Uygulama programında kullandığım (3.13.1) deyimindeki SENDCMD makrosu, VTAM'a bir işletmen komutunun iletilmesini sağlamaktadır. İşletmen komutlarına cevap olan VTAM mesajlarının alınması için (3.13.2) deyimindeki RCVCMD makrosunu kullandım. SENDCMD ve RCVCMD makroları, RPL-tabanlı, zaman-uyumlu işlemler talep etmektedirler.

```
SENDCMD  RPL=RPLSPO,                        (3.13.1)
         AREA=OCMD,
```

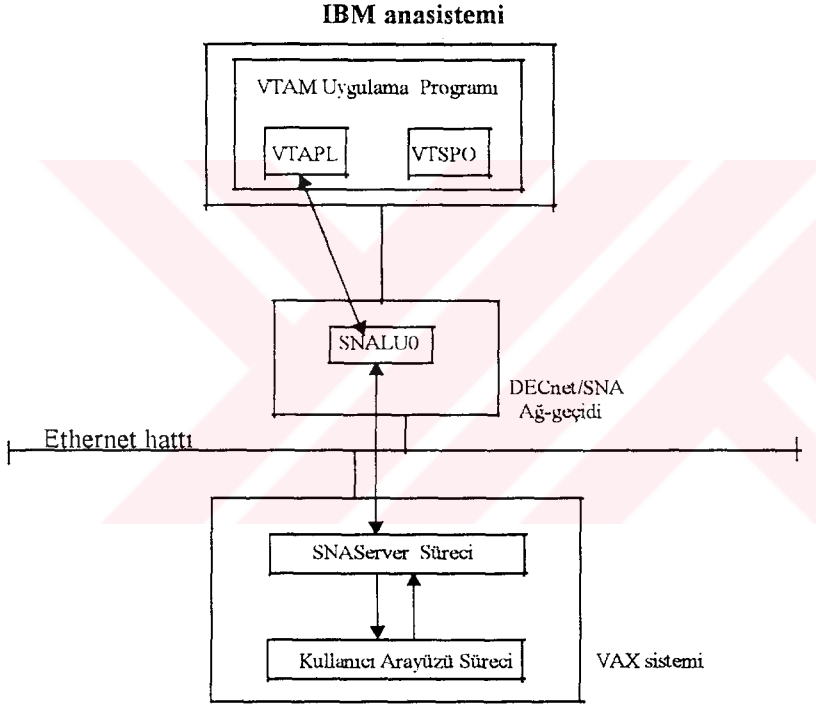
```
            RECLEN=OCMDLEN,
            OPTCD=SYN

RCVCMD      RPL=RPLSPO,                          (3.13.2)
            OPTCD=(SYN,NQ),
            AREA=ICMD,
            AREALEN=ICMDLEN
```

BÖLÜM 4

SNAServer ve Kullanıcı Arayüzü programlarını, VAX sisteminde bulunan VMS işletim sistemi altında C programlama dilinde geliştirdim. VMS işletim sistemi, uygulama programlarının geliştirilmesi için zengin bir ortam sağlayan, çoklu programlama özelliğine sahip bir işletim sistemidir. SNAServer ve Kullanıcı Arayüzü programları farklı süreçlerde çalıştırılmaktadır. SNAServer programı aynı ismi taşıyan bir süreç altında çalıştırılırken, Kullanıcı Arayüzü programı bir kullanıcı sürecinde çalıştırılmaktadır. Bu program birden fazla kullanıcı tarafından kullanılabilir.

4.1 SNAServer



Şekil 4.1 SNAServer için LU tanıtımı

SNAServer programı, DECnet/SNA ağ-geçidi altında tanımlı bir LU tarafından temsil edilmekedir. Bunu sağlamak için, DECnet/SNA ağ-gecidinde (4.1.1) deyiminde görülen SNALU0 erişim-noktasını tanımladım.

Access name = SNALU0 *(4.1.1)*
PU = SNA-0
LU list = 26
Logon mode = LUZERO
Application = VTAPL

SNAServer programınını SNA ağında temsil eden LU'nun ismi LUSERV'dir. Bu LU'nun VTAM tanımı (4.1.2) deyimindeki gibidir.

LUSERV LU LOCADDR=26, ISTATUS=ACTIVE, DLOGMOD=LUZERO (4.1.2)

LUZERO MODEENT LOGMODE=LUZERO,FMPROF=X'03',TSPROF=X'03' (4.1.3)
PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'4020',
PSERVIC=X'000000000000000000000000',RUSIZES=X'88',
PSNDPAC=X'08',SRCVPAC=X'04'

SNAServer ve VTAM Uygulama Programları arasındaki veri iletişimi, LUSERV ve VTAPL kavramsal birimleri arasında kurulan LU-LU oturumu üzerinden gerçekleşmektedir. Bu LU-LU oturumunda tip 0 protokolleri kullanılmaktadır. Bu oturum ile ilgili protokol bilgilerini belirtmek amacıyla, (4.1.5) deyiminde görülen LUZERO ismindeki 'logmode entry' sini tanımladım.

LU-LU oturumun kurulması, sonlandırılması ve bu oturum üzerinden veri iletişimin gerçekleştirilmesi için SNAServer programında, DECnet/SNA API fonksiyonlarını kullandım. DECnet/SNA API, tip 0 LU-LU oturumları için gerekli SNA protokol özelliklerini destekleyen fonksiyonlardan oluşmaktadır.

DECnet/SNA API fonksiyonlarını kullanarak aşağıdaki işlemlerin gerçekleşmesi sağlanır:
- Bir IBM uygulaması ile LU-LU oturumun kurulması talep edilir
- Bir IBM uygulamasına SNA İstek Birimleri gönderilir
- Bir IBM uygulamasından SNA İstek Birimleri alınır
- SNA Yanıt Birimleri gönderilir veya alınır
- Bir LU-LU oturumun sonlandırılması talep edilir.

Geliştirdiğim SNAServer programında aşağıdaki DECnet/SNA API fonksiyonları bulunmaktadır:

1. SNALU0$REQUEST_CONNECT

Bu fonksiyonu, SNAServer'ı temsil eden LU ve VTAPL arasında bir LU-LU oturumun kurulması için kullandım. Bu fonksiyon aşağıdaki işlemleri gerçekleştirmektedir:
- SSCP'ye bir INITSELF İstek Birimini göndererek LU-LU oturumun kurulmasını talep eder
- SSCP'nin gönderdiği INITSELF Yanıt Birimini alır
- VTAPL'den gelen BIND İstek Birimini alır
- Alınan BIND İstek Birimine karşılık olarak bir BIND Yanıt Birimi'ni gönderir.

2. SNALU0$REQUEST_DISCONNECT

Bu fonksiyonu, VTAPL ile kurulmuş olan LU-LU oturumun sonlandırılmasını sağlamak için kullandım.

Bu fonksiyon aşağıdaki işlemleri gerçekleştirmektedir:

- SSCP'ye bir Terminate İstek Birimini gönderek LU-LU oturumun sonlandırılmasını talep eder
- SSCP'den gelen Terminate Yanıt Birimini alır
-VTAPL'den gelen UNBIND İstek Birimini alır
-Bu İstek Birimine karşılık bir BIND Yanıt Birimini gönderir

3. SNALU0$TRANSMIT_MESSAGE

Bu fonksiyonu VTAPL` ye İstek Birim`lerin gönderilmesi için kullandım.

4. SNALU0$TRANSMIT_RESPONSE

Bu fonksiyonu, VTAPL`ye Yanıt Birim`lerin gönderilmesi için kullandım.

5. SNALU0$RECEIVE_MESSAGE

Bu fonksiyonu, VTAPL`nin gönderdiği İstek/Yanıt Birim`lerinin alınması için kullandım.

4.2 Süreçler arası iletişim

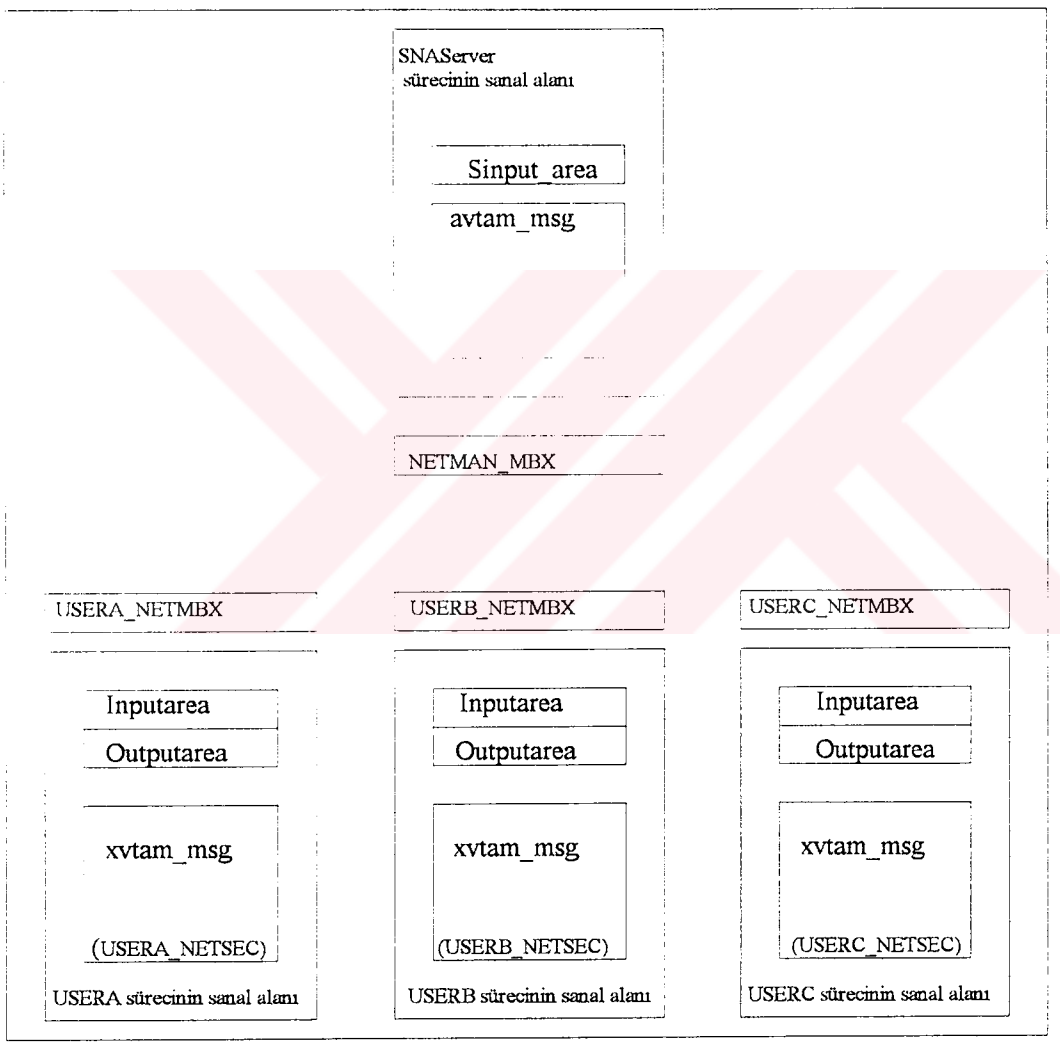
VMS süreçler arası iletişim için gerekli araç ve fonksiyonları sağlamaktadır. SNAServer ve Kullanıcı Arayüzü programlarının bulunduğu süreçlerin arasında veri iletişimi gerçekleştirebilmek için kullandığım araçlar, kavramsal-isimleri, posta-kutuları ve genel-bölümlerdir.

Bir **kavramsal-isim**, bir aygıt ismini veya uygulamaya özgü bir bilgiyi temsil eden ve kullanıcı tarafından belirtilen bir karakter dizilimidir. Bir eşdeğer-isim ise, asıl aygıt ismi veya karakter dizilimini ifade eden bir karakter dizilimidir. Bir kavramsal-isim tablosu, kavramsal-isim eşdeğer-isim çiftleri içeren, bağımsız bir isim alanıdır.

Posta-kutuları, süreçler arası veri iletişimini sağlamak için kullanılan sanal giriş/cıkış aygıtlarıdır. Veri transferi ise giriş/çıkış servislerini kullanarak gerçekleşir. Bir posta-kutusu üzerine giriş/cıkış işlemleri gerçekleştirebilmek için, önce bu aygıta bir kanal numarası atanmalıdır.

Bir posta-kutusu SYS$CREMBX sistem servisi ile yaratılır. Bu servis, yarattığı posta-kutusuna belirtilen kavramsal-ismi verir ve ona bir eşdeğer-isim atar. Ayrıca, SYS$CREMBX sistem servisi, yaratan süreç tarafından kullanılması için, posta-kutusuna bir kanal numarasını atamaktadır. Diğer süreçler ise, bu posta-kutusuna bir kanal numarasını atayabilmek için SYS$ASSIGN sistem servisini kullanmalıdırlar. SYS$ASSIGN servisinde, posta-kutusunu belirtmek için kavramsal-isimi kullanılmaktadır. Süreç, posta-kutusuna erişmeye artık gerek duymuyorsa, daha önce ona atadığı kanal numarasını serbest bırakmalıdır. Bunun için, SYS$DASSGN sistem servisi kullanır.

Bir aygıt üzerine giriş/çıkış işlemleri gerçekleştirmek için, SYS$QIO sistem servisi kullanılır. Bu serviste gerekli argümanlardan biri, giriş/çıkış işleminin yapılacağı aygıta önceden atanan bir kanal numarasıdır. Bir diğer argüman ise, gerçekleşmesi istenilen işlemi belirten işlev kodudur. SYS$QIO servisi ile talep edilen işlem zaman-uyumsuz bir işlemdir; talep kuyruğa eklenir eklenmez kontrol programa dönecektir. SYS$QIO servisinin zaman-uyumlu versiyonu SYS$QIOW dir. SYS$QIOW servisi, talep edilen işlemi kuyruğa ekler (queue eder) ve işlem tamamlanıncaya kadar programı bekletir. Bu tezdeki programlarda, posta-kutularına okuma/yazma işlemlerinin gerçekleştirilmesi için SYS$QIOW sistem servisini kullandım.



Şekil 4.2 : Süreçlerin sanal alanları

Genel-bölümler, iki veya daha fazla sürece ait ortak verilerinin geçici olarak saklanması için kullanılmaktadır. Genel-bölümün bir kopyası fiziksel bellekte bulunur. Onu paylaşan her süreç aynı

kopyaya başvurur. Bir süreç, bir genel-bölümüne erişebilmek için, önce bu genel-bölümünün, kendi sanal adres alanına eşlenmesini ('map' edilmesini) sağlamalıdır.

Bir süreç, bir genel-bölümünü yaratmak için SYS$CRMPSC sistem servisini kullanmalıdır. Bu servis ile, yaratılan genel-bölüm, yaratan sürecin sanal adres alanındaki bir bölgeye eşlenir. Diğer süreçler, bu genel-bölümü kendi sanal adres alanlarına eşlemek için SYS$MGBLSC sistem servisini kullanmalıdırlar.

Şekil 4.2'de kullanılan posta-kutuları ve genel bölümlerini içeren, kullanıcı ve SNAserver süreçlerinin sanal alanları görünmektedir.

Kullanıcı Arayüzü Programı'nın çalışma adımları:

1. Program, SNAServer sürecinin sistemde çalışıp çalışmadığını kontrol eder. SNAServer süreci çalışmıyorsa, program, kullanıcıya bir hata mesajı vererek sona erer.
2. Program, bulunduğu sürecin ismini öğrenir. Varsayalım bu sürecin ismi 'USERA' dır.
3. Program, SYS$CREMBX sistem servisini kullanarak 'USERA_NETMBX' kavramsal-ismini taşıyan bir posta-kutusunu yaratır.

SYS$CREMBX('USERA_NETMBX',xuser_kanal)

Bu işlemin sonucunda, posta-kutusuna atanan kanal numarası <xuser_kanal> değişkeninde bulunacaktır..

4. Program, SYS$ASSIGN sistem servisini kullanarak, SNAServer sürecinin yarattığı, 'NETMAN_MBX' ismindeki posta-kutusuna bir kanal numarasının atanmasını sağlar.

SYS$ASSIGN('NETMAN_MBX',xserver_kanal)

Atanan kanal numarası <xserver_kanal> değişkeninde bulunacaktır.

5. Program, SYS$CRMPSC servisi ile, 'USERA_NETSEC' isminde bir genel-bölüm yaratır . Yaratılan genel-bölüm, program sürecinin <xvtam_mesaj> ismindeki sanal alanına eşlenir.

SYS$CRMPSC('USERA_NETSEC',xvtam_mesaj,128)

6. Program, kullanıcının NCCF ortamından girdiği veya SNA ağ kaynaklarının durumunu izleme ekranlarının gerektirdiği, VTAM işletmen komutunu <Outputarea> alanına yazar. Ayrıca, bu alana kullanıcı genel-bölümünün ismini ('USERA_NETSEC') ve kullanıcı posta-kutusunun ismini ('USERA_NETMBX') ekler.
7. SYS$QIOW sistem servisi ile, <Outputarea> alanındaki veriler 'NETMAN_MBX' ismindeki posta-kutusuna aktarılır.

SYS$QIOW(xserver_kanal,IO$WRITE,Outputarea)

8. SYS$QIOW servisi ile, 'USERA_NETMBX' posta-kutusundan bir okuma işlemi başlatılır ve program, işlem tamamlanıncaya kadar bekletilir.

SYSSQIOW(xuser_kanal,IO$READ,Inputarea),

Bu işlemin sonucunda, posta-kutusundan, <Inputarea> alanına bir mesaj transfer edilmiş olur. <Inputarea> alanında, SNAServer sürecinin ilettiği ve istenilen işlemin tamamlandığını belirten bir mesaj bulunacaktır.

9. 'USERA_NETSEC' ismindeki genel-bölümün eşlendiği <xvtam_mesaj> alanında, SNAServer sürecinin yazdığı VTAM işletmen mesajları bulunacaktır. Program bu mesajları işler, sonra 6'ıncı adımdaki işleme geçer.

SNAServer Programı'nın çalışma adımları:

1. Program, VTAM uygulama programı ile bir LU-LU oturumunu kurar.

2. Program, SYS$CREMBX servisini kullanarak, 'NETMAN_MBX' kavramsal-ismini taşıyan bir posta-kutusunu yaratır.

SYSSCREMBX('NETMAN_MBX',aserver_kanal)

Bu işlemin sonucunda, posta-kutusuna atanan kanal numarası <aserver_kanal> değişkeninde bulunacaktır..

3. SYS$QIOW sistem servisi ile, 'NETMAN_MBX' posta-kutusundan bir okuma işlemi başlatılır ve işlem tamamlanıncaya kadar program bekletilir.

SYSSQIOW(aserver_kanal,IO$READ,Sinputarea)

Bu işlemin sonucunda, posta-kutusundan <Sinputarea> alanına veri aktarılmış olur. <Sinputarea> alanında, Kullanıcı Arayüzü programının 7'inci adımında 'NETMAN_MBX' posta-kutusuna aktarılan veriler bulunacaktır. Bu veriler, VTAM işletmen komutunu, kullanıcı genel-bölümünün ismini ('USERA_NETSEC'), ve kullanıcı posta-kutusunun kavramsal-ismini ('USERA_NETMBX') içermektedir.

5. SYS$MGBLSC servisi ile, 'USERA_NETSEC' ismindeki genel-bölüm, SNAServer sürecine ait bir sanal alanına eşlenecektir.

SYS$MGBLSC('USERA_NETSEC',avtam_mesaj)

Bu işlemin sonucunda, 'USERA_NETMAN' genel-bölümü, <avtam_mesaj> sanal alanına eşlenmiş olur.

6. Program, <Sinputarea> alanından alınan VTAM işletmen komutunu, VTAM uygulama programına gönderir. Daha sonra, VTAM uygulamasının gönderdiği, VTAM işletmen mesajlarını alır ve 'USERA_NETSEC' genel-bölümünün eşlendiği <avtam_mesaj> alanına yazar.

7. Program, SYS$ASSIGN servisi ile, 'USERA_NETMBX' posta-kutusuna bir kanal numarasını atar.

SYS$ASSIGN('USERA_NETMBX',auser_kanal)

Bu işlemin sonucunda posta-kutusuna atanan kanal numarası <auser_kanal> değişkeninde bulunacaktır.

8. Program, SYS$QIOW sistem servisi ile, 'USERA_NETMBX' posta-kutusuna 'işlem tamamlandı' mesajını yazar.

SYS$QIOW(auser_kanal,IO$WRITE, 'işlem tamamlandı')

9. Program, SYS$DASSGN servisi ile, 'USERA_NETMBX' posta-kutusuna atanan kanal numarasını serbest bırakır.

SYS$DASSGN(auser_kanal)

10. Program 3. adımdaki işleme geçer.

4.3 Kullanıcı Ekranları

Kullanıcı Arayüzü programı, iletişim sorumlusuna dört ana ekran sunmaktadır. Bunlar, NCCF, MAJNOD, LINES ve PUS ekranlarıdır.

1. NCCF - komut girme ekranı

NCCF, VTAM işletmen komutlarının giriş ekranıdır. Bu ekrandan, VTAM işletmen komutları girilmektedir ve bu komutlara cevap olan VTAM işletmen mesajları görüntülenir. Bilinen VTAM komutları dışında, CLIST'ler de kullanılabilir. CLIST komutları, komut listeleridir. CLIST'ler, sıkça kullanılan VTAM komutlarının, bir tek komutun girilmesi ile işlenmesine imkan vermektedir.

Örnekler:

D NET,ID=LINEA,E	komutu yerine *G LINEA*	CLIST'i girilebilir.
V NET,REL,ID=PUA	komutu yerine *REL PUA*	CLIST'i girilebilir
V NET,INACT,ID=PUA,F	komutu yerine I PUA	CLIST'i girilebilir

NCCF ortamında, birden fazla VTAM komutu ile eşdeğer CLIST komutları da girilebilmektedir. Kullanıcı, bir CLIST komutunu girmekle, iki veya daha fazla VTAM komutunun işlenmesini sağlar.

Örnekler:

INA LN425 komutu,

V NET,INACT,ID=PU425,F

V NET,INACT,ID=LN425,F

komutların yerine kullanılabilir.

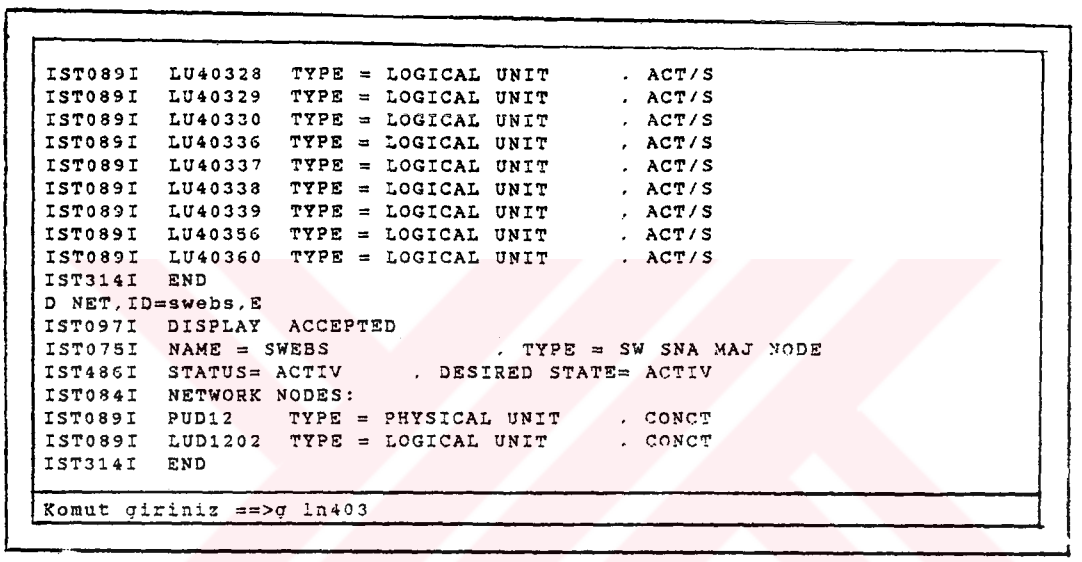
ACT LN425 komutu,

V NET,ACT,ID=LN425

V NET,ACT,ID=PU425,SCOPE=ALL,LOGON=CICSEBS

komutların yerine kullanılabilir.

Ayrıca NCCF ortamında, SNA ağındaki bütün LINE ve PU'ları etkin veya işlem-dışı durumuna getiren CLIST komutları kullanılabilmektedir.



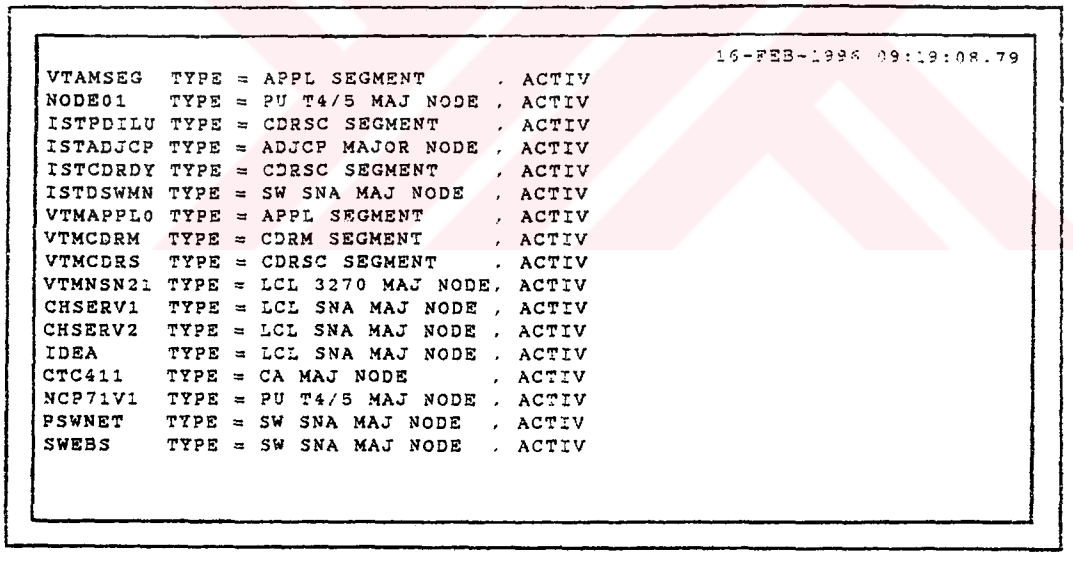
Şekil 4.3 : NCCF Ekranı

2.Kaynakların durumunu izleme ekranları

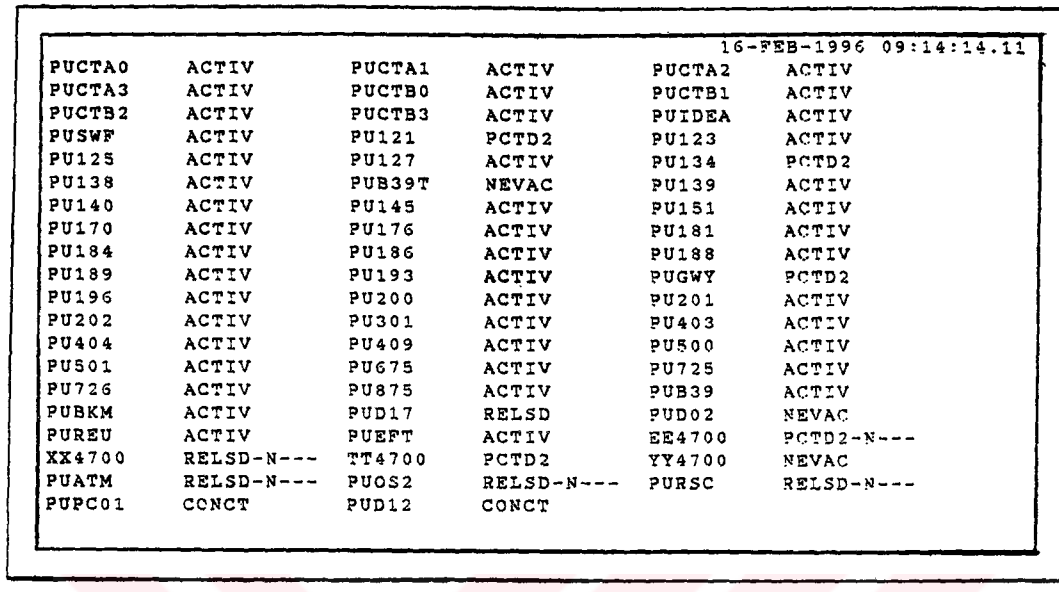
MAJNOD, LINES ve PUS ekranları ağ kaynaklarının durumunu izleme ekranlarıdır. Bu ekranlar, belli zaman aralıklarında kendilerini tazeleyerek, ağ kaynaklarının son durumunu gösterirler. MAJNOD ekranı, etki-alanında bulunan tüm Büyük Düğüm'lerin durumlarını görüntüler. LINES ekranı etki-alanında bulunan tüm hatların (LINE' ların) durumlarını izleme ekranıdır. PUS ekranı ise, etki-alanında bulunan tüm Fiziksel Birimlerin (PU'ların) durumlarını izleme ekranıdır. Program, LINES ve PUS ekranlarını tazelerken, işlem-dışı durumunda olan kaynakları etkin durumuna getirmek için gerekli VTAM komutlarının işlenmesini sağlamaktadır. Bu şekilde, iletişim işletmeninin müdahalesine gerek duyulmadan kaynaklar etkin durumuna getirilmektedir. Durum izleme ekranlarından bir ağ kaynağı seçilebir ve fonksiyon tuşlarını kullanarak seçilen kaynak için, aşağıda belirtilen işlemler gerçekleştirilebilir.

Fonksiyon tuşları:

- DISPLAY tuşu ile kullanıcı, seçilen kaynak hakkında ayrıntılı bilgi edinebilir ve bu kaynağın altında bulunan alt kaynaklarının durumlarını gözleyebilir.
- INACT tuşu ile kullanıcı, seçtiği kaynağı işlem-dışı durumuna getirebilirr
- ACT tuşu ile kullanıcı, seçilen kaynağı etkin durumuna getirebilir.
- REL tuşu, PU'lar için geçerlidirler. Kullanıcı, bu tuş ile, *V NET,REL,ID=puname* VTAM komutunun işlenmesini sağlayabilir
- ACQ tuşu, PU'lar için geçerlidirler. Kullanıcı, bu tuş ile, *V NET,ACQ,ID=puname* VTAM komutunun işlenmesini sağlayabilir.



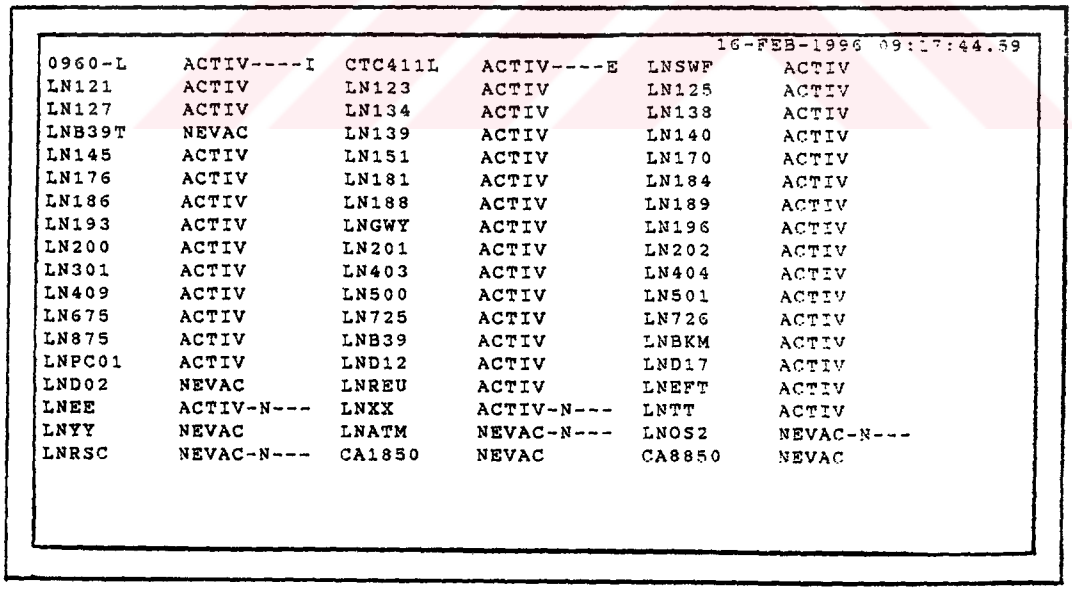
Şekil 4.4: MAJNOD (Büyük Düğüm'ler) ekranı



16-FEB-1996 09:14:14.11

PUCTA0	ACTIV	PUCTA1	ACTIV	PUCTA2	ACTIV
PUCTA3	ACTIV	PUCTB0	ACTIV	PUCTB1	ACTIV
PUCTB2	ACTIV	PUCTB3	ACTIV	PUIDEA	ACTIV
PUSWF	ACTIV	PU121	PCTD2	PU123	ACTIV
PU125	ACTIV	PU127	ACTIV	PU134	PCTD2
PU138	ACTIV	PUB39T	NEVAC	PU139	ACTIV
PU140	ACTIV	PU145	ACTIV	PU151	ACTIV
PU170	ACTIV	PU176	ACTIV	PU181	ACTIV
PU184	ACTIV	PU186	ACTIV	PU188	ACTIV
PU189	ACTIV	PU193	ACTIV	PUGWY	PCTD2
PU196	ACTIV	PU200	ACTIV	PU201	ACTIV
PU202	ACTIV	PU301	ACTIV	PU403	ACTIV
PU404	ACTIV	PU409	ACTIV	PU500	ACTIV
PU501	ACTIV	PU675	ACTIV	PU725	ACTIV
PU726	ACTIV	PU875	ACTIV	PUB39	ACTIV
PUBKM	ACTIV	PUD17	RELSD	PUD02	NEVAC
PUREU	ACTIV	PUEFT	ACTIV	EE4700	PCTD2-N---
XX4700	RELSD-N---	TT4700	PCTD2	YY4700	NEVAC
PUATM	RELSD-N---	PUOS2	RELSD-N---	PURSC	RELSD-N---
PUPC01	CONCT	PUD12	CONCT		

Şekil 4.5: PUS (Fiziksel Birim'ler) ekranı



16-FEB-1996 09:17:44.59

0960-L	ACTIV----I	CTC411L	ACTIV----E	LNSWF	ACTIV
LN121	ACTIV	LN123	ACTIV	LN125	ACTIV
LN127	ACTIV	LN134	ACTIV	LN138	ACTIV
LNB39T	NEVAC	LN139	ACTIV	LN140	ACTIV
LN145	ACTIV	LN151	ACTIV	LN170	ACTIV
LN176	ACTIV	LN181	ACTIV	LN184	ACTIV
LN186	ACTIV	LN188	ACTIV	LN189	ACTIV
LN193	ACTIV	LNGWY	ACTIV	LN196	ACTIV
LN200	ACTIV	LN201	ACTIV	LN202	ACTIV
LN301	ACTIV	LN403	ACTIV	LN404	ACTIV
LN409	ACTIV	LN500	ACTIV	LN501	ACTIV
LN675	ACTIV	LN725	ACTIV	LN726	ACTIV
LN875	ACTIV	LNB39	ACTIV	LNBKM	ACTIV
LNPC01	ACTIV	LND12	ACTIV	LND17	ACTIV
LND02	NEVAC	LNREU	ACTIV	LNEFT	ACTIV
LNEE	ACTIV-N---	LNXX	ACTIV-N---	LNTT	ACTIV
LNYY	NEVAC	LNATM	NEVAC-N---	LNOS2	NEVAC-N---
LNRSC	NEVAC-N---	CA1850	NEVAC	CA8850	NEVAC

Şekil 4.6: LINES (Hatlar) ekranı

SONUÇLAR

Tez çalışmam üç uygulamadan oluşmaktadır. Bunlar IBM anasisteminde geliştirdiğim VTAM Uygulaması ve VAX sisteminde geliştirdiğim SNAServer ve Kullanıcı Arayüzü uygulamalarıdır. IBM anasisteminde assembler dilinde bir uygulama geliştirmenin zorluğu, 'debug' ve 'trace' olanaklarının olmaması, SNA'de veri alış-verişinin teferruatlı ve karmaşık olması, VTAM uygulamasını geliştirirken karşılaştığım zorluklardır. VMS işletim sisteminin sunduğu olanaklar ve C programlama dilini kullanmam sayesinde, VMS altındaki programları geliştirmek nispeten daha kolay oldu.

Bu tez calışmasında yazdığım uygulamalar geliştirilmeye açıktır. Örneğin, VAX sisteminde, istatistiksel bilgi toplayan, SNAServer süreci ile veri iletişiminde bulunacak, yeni bir program yazılabilir. Bunun için VTAM uygulama programına bir PPO eklenmeli. PPO (Primary Program Operator), bir Birincil Program İşletmenidir. PPO, VTAM uygulama programını temsil edebilecek ücüncü bir Uygulama LU'sudur.

Diğer bir önerim, VAX ortamında bulunan programları bir PC üzerinde geliştirmektir. Bunun için PC ve IBM sistemi arası iletişimi sağlamak amacıyla bir SNA kartına ve SNA/API'ye gerek duyulacaktır. VTAM uygulama programı da buna uygun şekilde uyarlanacaktır. PC üzerinde geliştirilecek uygulamaya GUI (graphical user interface) dahil edilebilir.

KAYNAKLAR

1 - GC30-3073-2 - Systems Network Architecture - Technical Overview

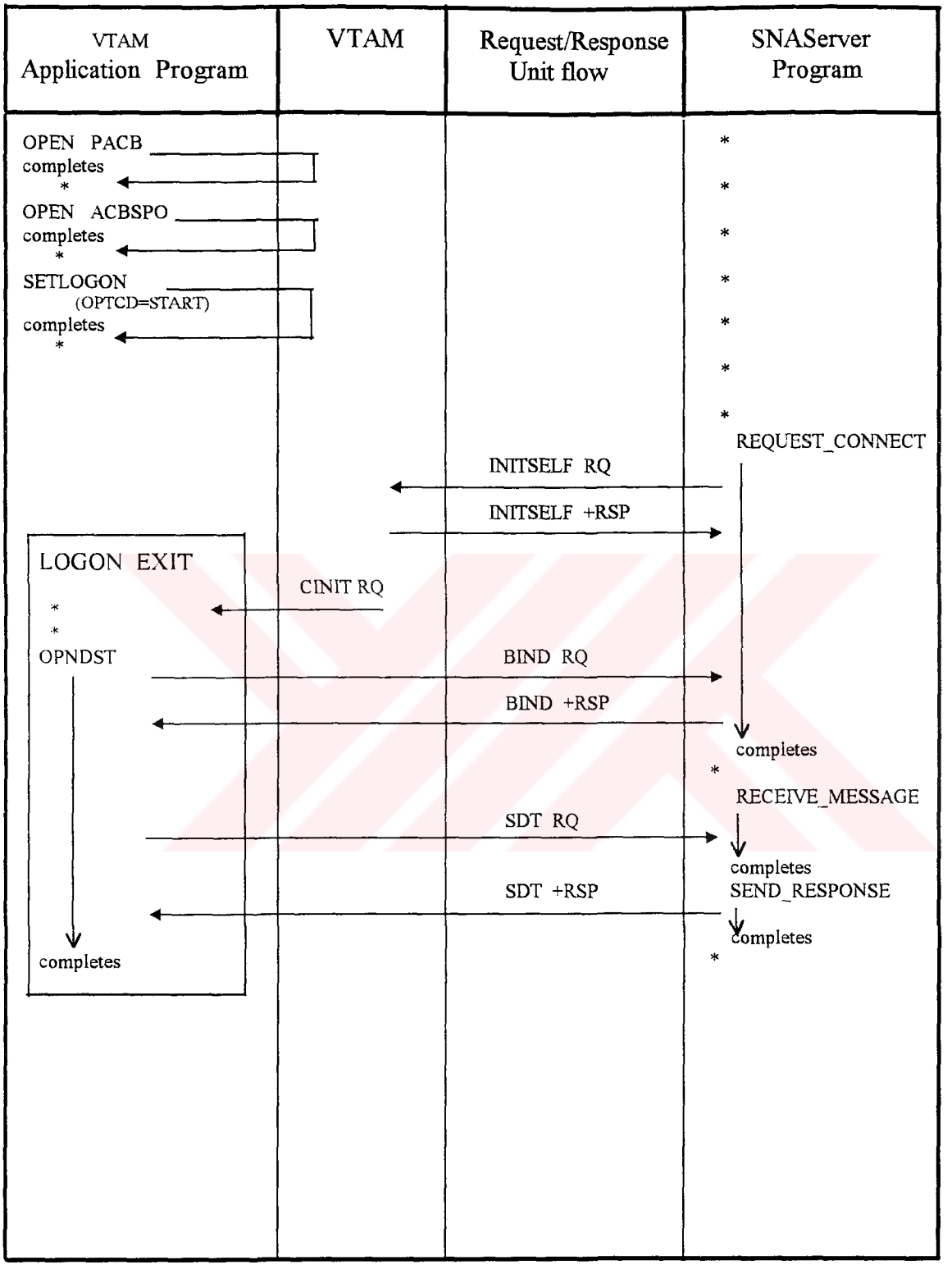
2 - SC31-6436-1 - VTAM Programming

3 - SC31-6493-01 - VTAM Messages and Codes

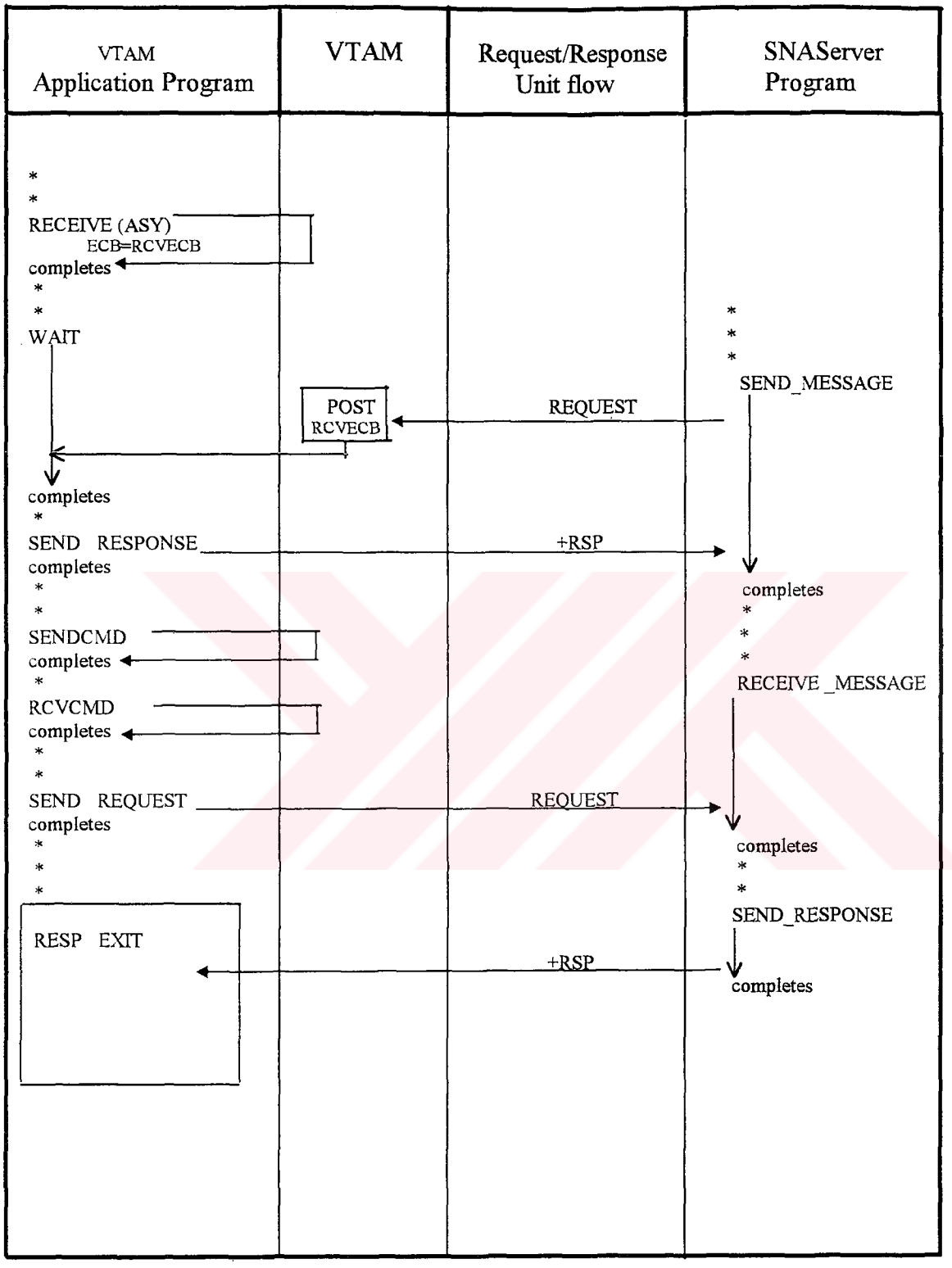
4 - Frank, Carrano - Assembler Language Programming for the IBM 370

5 - Digital Equipment Corporation - OpenVMS Programming Concept Manual

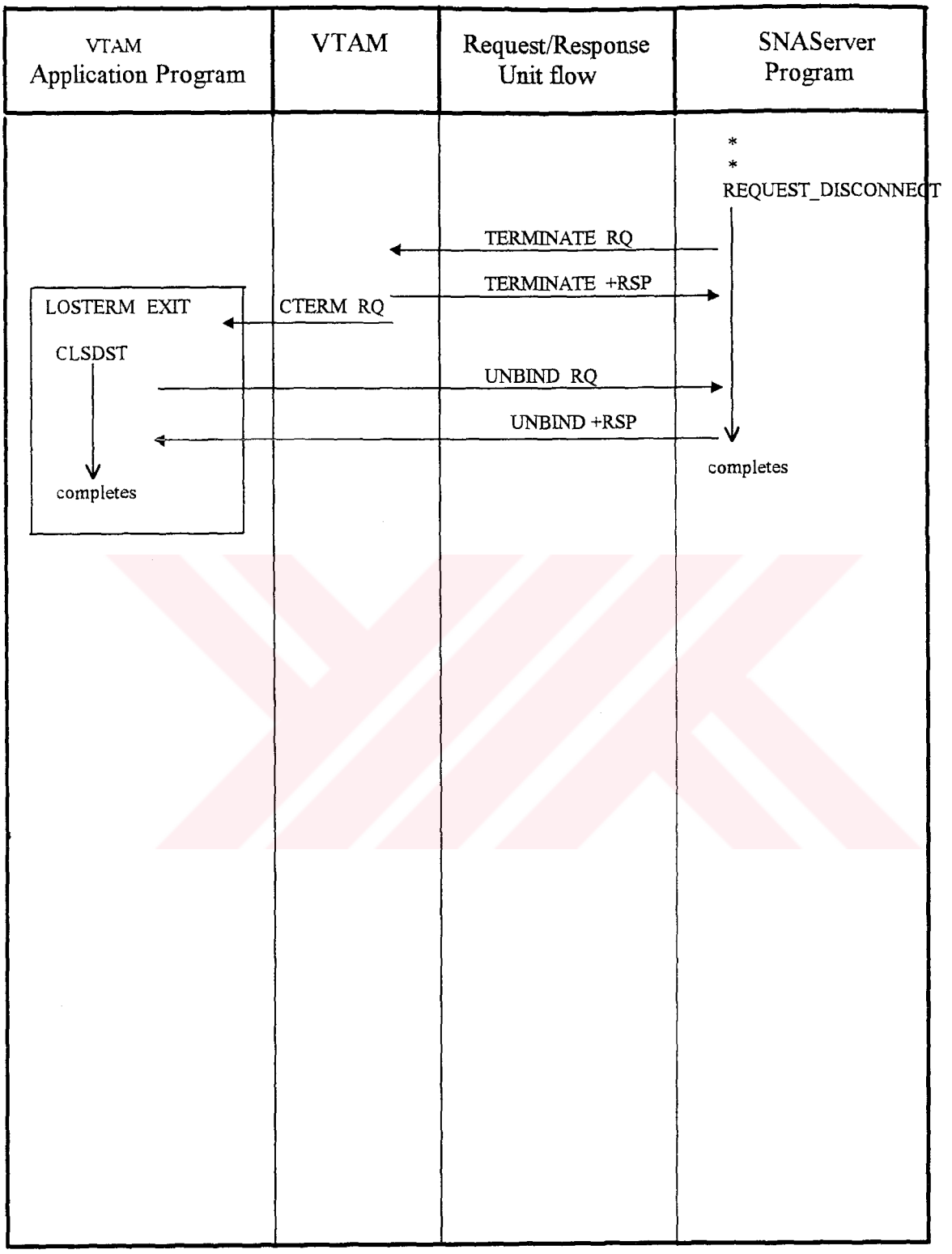
6 - Digital Equipment Corporation - OpenVMS System Services



EK 1: LU-LU oturumun kurulması



EK 2: LU-LU oturumu üzerinden veri alış-verişi



EK 3: LU-LU oturumun sonlandırılması

ÖZGEÇMİŞ

Doğum Tarihi : 08.12.1967

Doğum Yeri : Köstence , Romanya

1974 - 1978 : G.O. 23 , Köstence, Romanya

1978 - 1982 : G.O. 23 , Köstence, Romanya

1982 - 1986 : ' Mircea cel Batrın' Lisesi, Köstence

1989 - 1993 : YTÜ Elektrik Elektronik Fakültesi, Bilgisayar Bilimleri ve Mühendisliği Bölümü

1993 - : YTÜ Fen Bilimleri Enstitüsü, Bilgisayar Bilimleri ve Mühendisliği Anabilim dalı