

T. C.
EGE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

BİLGİSAYAR GÖRÜNTÜ MAKİNELER İŞLETİM SİSTEMİNDE
TOPLU İŞ GİRİŞİ UYGULAMALARINA ARABİRİM TASARIMI
VE
GERÇEKLEŞTİRİMİ

Hazırlayan : Cengizhan ALIÇLI

Danışman : Doç. Dr. Mesut RAZBONYALI

YÜKSEK LİSANS TEZİ

BORNOVA - İZMİR
1989

Ö N S Ö Z

Günümüzde tüm sistemlerin gerçek zamanlı ve çok kullanıcıya olma yolunda çok hızlı ve sıkı adımlarla ilerlediği bilinmektedir. Bunun yanında sistem performanslarının ve buna bağlı olarak da sistemde çalışan ve çalışması düşünülen yazılımların, neler getirip neler götüreceği çok iyi bir şekilde incelenmektedir. Sistemler bu yolda ilerlerken, sistem mantığına bağlı olarak ortaya çıkmış olan toplu işletimler de hala kullanılır durumdadır. Ancak bunun yanında, batch işletim sistemlerinin tarihe karışmak üzere olduğunu kabul etmekte de yarar bulunmaktadır. Tüm bunlara karşın sistemlerde hala çalışmakta olan bu tür yazılımların, kullanıcılara bazı kolaylıklar getirdiği de tartışılmaz bir gerçektir.

Çalışmalarında yardım ve desteklerini esirgemeyen, hocam, Prof.Dr.Oğuz MANAS'a en içten teşekkürlerimi sunarım.

Konuda beni çalışma yapmam için yönlendiren, teşvik eden ve destekleyen tez danışmanım Doc.Dr. Mesut RAZBONYALI'ya, ayrıca çalışmam sırasında döküman araştırma, inceleme ve takibinde bana sonsuz destek veren Dr. Murat AŞIKHAN'a ve tezimin yazımında benden hiç bir yardımı esirgemeyen değerli arkadaşlarım Ar.Gör Ferzan YALKUT, Ar.Gör Reha ÇETİN ve Ar.Gör Levent TOKER'e teşekkür etmeyi bir borç bilirim.

Bugünlere gelmemde hiç bir fedakarlıktan kaçınmayan, beni her zaman destekleyen aileme ve eşime olan şükran borcumu hiçbir şekilde ödeyemem.

Cengizhan ALIÇLI

1989

Dear Joan and Oğuz,

This project certifies that from now on you are dealing with a brother who has got his M.S. degree in Computer Engineering. I hope and want my brother will do the same and more better. I hope I can sent my Ph.D. project too.

I love you both. Cengizhan (shrestha, cda)

İ Ç İ N D E K İ L E R

SAYFA

ÖNSÖZ	
İÇİNDEKİLER	1
ŞEKİL LİSTESİ	4
ABSTRAKT	5
ABSTRACT	5
1 - GİRİŞ	6
2 - MATERYAL VE METOD	10
3 - ONLINE VE BATCH SİSTEMLER	13
3.1 - PROCESS SEÇME	14
3.2 - ONLINE SİSTEMLER	17
3.3 - BATCH SİSTEMLER	20
4 - VARSAYILAN BELLEK	23
4.1 - VARSAYILAN BELLEK NEDİR ?	23
4.2 - VM / SP İŞLETİM SİSTEMİ HAKKINDA BİLGİ	26
4.3 - CP HAKKINDA BİLGİ	29
4.4 - CMS HAKKINDA BİLGİ	30
5 - CMSBATCH NEDİR ?	33
5.1 - CMSBATCH ÖZELLİĞİNİ KULLANMAK	33
5.2 - VMBATCH SİSTEMİ NEDİR ?	36
5.2.1 - VMBATCH ALT SİSTEMİNİN KULLANILMASI	37
6 - SİSTEM PROGRAMLAMA AÇISINDAN CMSBATCH ÖZELLİĞİ	39
6.1 - CMSBATCH MAKİNASİNİN TANIMLANMASI	40
6.2 - CMSBATCH MAKİNASİNİN SİSTEM LİMİTLERİNİN DEĞİŞTİRİLMESİ	42
6.3 - CMSBATCH MAKİNASINA ÖZEL BAZI DURUMLARI KONTROL ETMEK İÇİN ALTYORDAMLAR YAZMAK	43

	SAYFA
6.3.1 - BATEXIT1 ALTYORDAMI	43
6.3.2 - BATEXIT2 ALTYORDAMI	44
6.4 - BATCH ÖZELLİĞİNDE VERİ GÜVENLİĞİ	45
6.5 - SAKLANMIŞ SİSTEM KULLANMAKLA IPL PERFORMANSININ ARTTIRILMASI	46
7 - CMSBATCH MAKİNESİNİN OPERATÖR İLE İLİŞKİSİ	47
7.1 - BATCH MAKİNASININ ÇALIŞTIRILMASI	48
7.2 - BATCH MAKİNASININ KONTROL EDİLMESİ	49
7.3 - BATCH İŞLERİN SIRALANMASI , SİLİNMESİ VE BAŞLATILMASI	51
7.4 - BATCH MAKİNASININ DURDURULMASI	53
8 - KULANICILAR AÇISINDAN BATCH MAKİNASI	54
8.1 - BATCH MAKİNASINA İŞ YOLLAMAK	54
8.2 - BATCH MAKİNASINA YOLLANACAK İŞ'DE OLMASI GEREKLİ ÖZELLİKLER	55
8.3 - BATCH MAKİNASINA VARSAYILAN KART GİRİŞİ	56
9 - BATCH MAKİNASININ ÇALIŞMA ÖZELLİĞİ	61
9.1 - BATCH İŞLEMİN HAZIRLANMASI	62
9.2 - BATCH İŞLEMDE CP VE CMS KOMUTLARINDAKİ KISITLAMALAR	64
9.3 - BATCH ÇIKTILARI	66
10 - GERÇEKLEŞTİRİLEN SİSTEMİN TANITIMI	68
10.1 - SİSTEMİN GENEL TANITIMI	68
10.2 - SİSTEMİN TANITIMI	69
10.2.1 - SİSTEMİN ÇALIŞTIRILMASI	71
10.2.2 - MENÜLER ve AÇIKLAMALARI	72
10.2.2.1 - BATCH0 MENÜSÜ	73
10.2.2.2 - BATCH1 MENÜSÜ	73
10.2.2.3 - BATCH2 MENÜSÜ	75
10.2.2.4 - BATCH3 MENÜSÜ	76
10.2.3 - KULLANILAN KÜTÜK AÇIKLAMALARI ve AMAÇLARI	78
10.2.3.1 - FILE USER KÜTÜĞÜ	78
10.2.3.2 - CM EXEC KÜTÜĞÜ	78
10.2.3.3 - FILE HELP KÜTÜĞÜ	79

	SAYFA
10.2.4 - KULLANILAN EXEC KÜTÜKLERİNİN AÇIKLANMASI	80
10.2.4.1 - CM2 EXEC	80
10.2.4.2 - BATYIL EXEC	81
10.2.4.3 - YILDIZ EXEC	81
11 - SONUÇ	82
ÖZET	83
SUMMARY	84
LİTERATÜR LİSTESİ	85
EKLER	86

SEKIL LISTESI

	SAYFA	
3.1	OLRT ÇALIŞMA ŞEKLİNE BİR ÖRNEK	19
3.2	BATCH ÇALIŞMADA OKUYUCU FORMATI	21
3.3	BATCH ÇALIŞMAYA ÖRNEK BİR AKIŞ	22
4.1	DEMAND PAGE ALLOCATION GÖSTERİMİ	24
4.2	CP ORTAMINA BİR ÖRNEK	29
4.3	CP ORTAMINDA AYRICALIK DERECELERİ	29
4.4	VM İŞLETİM SİSTEMİ MANTIĞINA ÖRNEK	32
10.1	FILE USER KÜTÜK FORMATI	78
10.2	CM EXEC KÜTÜK FORMATI	79
10.3	FILE HELP KÜTÜK FORMATI	80

ABSTRAKT

Bu çalışmada bilgisayar sistemlerinde her zaman var olabilecek batch işletimler ile iletişim kurulmasındaki zorluklar araştırılmıştır. Bu zorluklara çözüm olarak bir arabirim tasarlanmış ve gerçekleştirilmiştir. Arabirimin tasarımında ve gerçekleştiriminde PL/I programlama dili, kullanıcı ile olan karşılıklı iletişimi sağlamak için yardımcı olarak DMS(Display Management System) paketi kullanılmıştır.

Projede kullanılan ortam, Ege Üniversitesi Bilgisayar Merkezi bilgisayar sistemi olup, bu ortamda CMSBATCH makinası ile çalışılmıştır. Meydana gelen sorunlar irdelenmiş ve kullanıcıların rahatça yararlanabileceği bir program yapılmaya çalışılmıştır.

ABSTRACT

In this project, we worked on the difficulties in communication between the batch systems, because of JCL's(Job Control Language). For a solution to this problem, an interface program had been developed. During this studies, PL/I was used as a programming language and besides, in helping the communication via the interactive screens, DMS was also an assistant as a utility.

During the project, we studied on the computer of the Ege Üniversitesi Bilgisayar Merkezi, and mostly CMSBATCH virtual machine. The problems that were faced was discussed briefly. As a result, we tried to develop a program which is useful and easy to use.

1 - GİRİŞ

Günümüz bilgisayar dünyasında meydana gelen ileriye dönük gelişmeler günümüzde herkezi büyülemektedir. Örneğin konuşan robotlar, hiçbir canlının yapamayacağı kadar hassas ölçümlmeler yapma gibi özelliklere sahiptirler. Tüm bunların yanında, daima daha hızlı ve kapasitesi daha fazla bilgisayar üretme çalışmaları günümüzde daha da hız kazanmıştır. Bu çalışmaların sonucu olarak, karşımıza süper bilgisayarlardan, üretimi artık yapılmayan bilgisayarlara kadar çeşitli cins, marka ve modelde bilgisayarlar çıkmaktadır.

Tüm bu çeşitliliklerine rağmen kullanıcılarına sağladıkları olanaklar, kullanıcılar açısından düşünüldüğünde o kadar da farklılık göstermemektedir. Ancak, bilgisayarlar açısından düşünüldüğü zaman, yine çok fazla olmamakla birlikte dış dünya ile ilişkilerini sağlayan ve kullanıcılarına en az yük bırakmaya çalışan çeşitli yazılımlar vardır ki, bunlara **İşletim Sistemi** denilmektedir. Bazı sistemler sadece bu yazılımlara verilen adlarla isimlendirildikleri halde, bu yazılımlar arasındaki çeşitlilik, bilgisayarlar arasındakinden daha azdır. Ancak,

tüm dünyada yapıla gelen bir çalışma da, programlama dillerinde olduğu gibi, ortak ve tüm sistemlerde çalışabilecek bir işletim sistemi tasarımıdır.

Günümüzde var olan sistemlerde çalışmakta olan işletim sistemleri arasında, Time Sharing, Multiprocessing, Batch Processing, Multiprogramming, Remote Batch Processing, Online, Interactive, Conversational, Sequential Processing, Parallel Processing sayılabilir.

Bu çalışmada, Online ve Batch işletim sistemleri ile çalışılmıştır. Merkezimizde kurulu bulunan IBM sisteminde çalışır durumda olan VM işletim sistemi multiprogramming ve time sharing bir sistemdir. Bu yüzden bilgisayarımızda hem online hem de batch uygulamalar aynı anda çalışabilmektedir.

Bilgisayar düşüncesinin ilk ortaya çıkmaya başladığı zamanlarda gerçekleştirilen ve hala kullanılan BATCH işlem ile, nisbeten daha yeni kullanılmakta olan ONLINE işlem arasındaki, var olan diğer en belirgin ayrıcalık, işlem zamanıdır. Herhangibir noktadan herhangi bir istek geldiği zaman, bu isteğin anında karşılanmaya çalışılması, online işlem mantığını oluşturur.

Batch işlemden ise, bu mantık daha basite indirgenmiştir.

Meydana gelebilecek olan herhangi bir istek, öncelikle sıraya konmakta ve bu istekten daha önce oluşmuş diğer istekler yanıtlanıncaya değin beklemeye alınmaktadır. Burada da açıkça anlaşıldığı gibi **BATCH** işleminde, **ONLINE** işleme göre daha fazla bekleme zamanı geçmektedir.

Bu yüzden **IBM 4341** sistemimizde oluşan batch işlerdeki darboğazın biraz da olsa önüne geçebilmek için, sistemde kullanılır durumda olan ancak kullanabilmek için mevcut komutlar dizisinden, yapılmak istenen işe bağlı olarak, çeşitli kombinasyonlar kullanımasının gerekliliği, kullanıcıları **CMSBATCH** olayından doğal olarak uzak tutmuştur. Felsefe olarak sistemde bir anda, beş yada altı batch işin bulunması yerine, bu batch işler kendi içinde ilk gelen, ilk servis alır mantığı ile sıralanır. Daha sonra sisteme birer birer verilip böylece de sistemde bir anda bir tane batch işin çalışmasını sağlamak, hem sistemin performansı hem de kullanıcılar açısından bir rahatlama nedeni olabilecektir. Bu düşünceyle **CMSBATCH** makinası ile kullanıcılar arasında bir arabirim oluşturulması tasarlanmıştır.

Tasarlanan bu sistemin ayrıntılı açıklanmasından önce, tartışılan BATCH ve ONLINE sistemler hakkında bilgi verilecek, daha sonra sistemimizde mevcut olan ve her iki tür çalışmayı mantığı gereği kolaylıkla sağlayabilen VM işletim sistemi ile ilgili ayrıntılı bilgi verilecektir. Son bölümde ise tasarımılanan sistemin tanıtımı, karşılaşılan güçlükler, akış şemaları ve kaynak programın kendisi bulunmaktadır.

2 - MATERYAL ve METOD

Günümüzde kurulu bulunan birçok bilgisayar sisteminde temel olan ortak arzu, sistemden maksimum faydayı minimum yük getirerek sağlamaktır. Bu yüzden bu isteğe bağlı olarak birçok sistem tasarlanmış ve yan birim geliştirilmiştir. Bazılarında tek CPU, bazılarında birden çok CPU, bazılarında ise bunların bir destekleyicisi olarak yan birimlerde birer CPU bulunmaktadır. Tek CPU 'lu olarak tasarımılanan sistemlere **MULTIPROGRAMMING** (ÇOKLU PROGRAMLAMA) ortamı bulunan sistemler, birden çok CPU'lu olarak tasarımılanan sistemlere ise **MULTIPROCESSING** (ÇOK İŞLEYİCİLİ) sistemler denilmektedir. Bazı durumlarda, sistemin çok yüklü olduğu durum gibi, meydana gelen herhangi bir isteği uzak bir CPU' da çözümleyip sorunu ortadan kaldırmak mantığı kullanılır, bu tür sistemlere ise **REMOTE CPU' lu sistemler** denilmektedir.

Yukarıda sözü edilen üç ayrı sistemde çalışması olası olan programlar, etkileşimli ve batch olarak çalışabileceklerdir. Burada çalışma prensibi zaman paylaşımli olan sistemlerden bahsetmek yerinde olacaktır. Bu tür sistemlerde, genelde çalışma düzenine bağlı olarak üç tür kuyruk bulunmaktadır. Mevcut sistemimizde

ise Q1, Q2, Q3, E1, E2 olmak üzere beş ayrı kuyruk bulunmaktadır. Q1 kuyruğu karşılıklı iletişimli programların servis verilmek üzere bekleyeceği, Q2 ve Q3 ise batch işlerin çalışacağı kuyruklardır.

Sistemimizde yapılan araştırmalara göre, ne zaman alınıralsa alınmış olsun, sistemin performansı ile ilgili bazı dökümanlar elde ettiğimizde mevcut olan sıkışıklığın büyük bir bölümünün batch kuyruklarında oluşmaktadır. Buna karşın, karşılıklı iletişimli işlerin çalışacağı kuyrukta ise hemen hemen hiçbir beklemenin olmadığı gözlenmiştir. Bunun nedeni, sistemimizde çalışmakta olan işlerin içeriğidir. Her ay sonuna doğru çalışmaya başlayan PTT ve TEK programları, bunların yanında çeşitli iterasyonlar yapan istatistik makinaları, aynı anda sistemden kaynak bekler duruma geldiği anda sistemin performansı buna bağlı olarak oldukça düşme göstermektedir. Bu duruma bir çözüm olarak CMSBATCH makinası öngörülebilir. Sistemde bir anda bu tip makinaların sadece bir tanesi çalışır olacağından, batch kuyruğundaki sıkışıklığın tamamen olmasada, azalacağı söylenebilir.

Bu düşünceden yola çıkarak, sistemimizde çalışmaya hazır durumda bulunan CMSBATCH yığıt makinasının kullanılabilmesi için, bir arabirim tasarımı ve gerçekleştirimi planlanmış ve bu çalışmanın konusu olmuştur. Sistemin tasarlanması sırasında VM / SP (Virtual Machine / System Product) Release 3 işletim sistemi bulunan IBM 4341 Model grup 2 bilgisayarı kullanılmıştır.

3 - ONLINE ve BATCH SİSTEMLER

İşletim sistemleri genel olarak üç ana grup altında toplanmaktadır (Alan C-Shaw, 1974) ;

- 1.- **Batch Processing (Yığıt İşleme).**: Bu tür sistemlerde, kullanıcının sisteme yolladığı iş, sistemin INPUT birimi aracılığıyla, bu birimden sırayla işlenir. İşletim sırasında kullanıcı ile bir iletişim kurulması söz konusu değildir. Kullanıcı tamamiyle yazdığı programdan çalışma sırasında uzak kalmakta, bundan dolayı da sistem, çalışan programın çalışma zamanı kadar olan süre içinde başka hiçbir işle meşgul olamamaktadır.
- 2.- **Time Sharing (Zaman Paylaşımı).**: İşleyici zamanının, sistemde var olan işler arasında paylaşılmasını, aynı zamanda kullanıcının yazdığı programla ilişki kurmasını sağlayan sistemlerdir. Bu olay, sistemin çalışma zamanının, kullanıcılar arasında paylaşılması ile sağlanmaktadır. Kullanıcıların çalışabilmesi için sağlanan bu zaman dilimi sınırında, yollanan iş bitmeyebilir. Bu durumda bu iş belli bir bekleme kuyruğuna alınır. Daha sonra işleyici

zamanı, ya bekleme kuyruğundaki başka bir işin çalışabilmesi için o işe verilir ya da kuyrukta olmayan ancak bir istekte bulunan bir işe tahsis edilir.

3.- Real Time.: Sisteme gelebilecek olan isteklerin nereden gelebileceği kesinlikle belli olan sistemlerdir. Sisteme gelen herhangi bir istek olduğu zaman, işleyici o anda yapmakta olduğu işi bırakır. Hangi birimden kesintinin geldiği bulunur ve diğer işi bitirene kadar o işe hizmet vermez. Yapmakta olduğu işi bitirdiği anda ise kesinti veren işe başlar. Burada, bu kontrol mekanizmasının çok iyi çalışması gerekmektedir. Aksi halde kesinti veren bir işin verdiği kesintiye cevap verilemez. Yani kesintinin olduğu anlaşılamıyabilir. Bu tür sistemler, genelde bir fabrika otomasyonu ya da büyük sanayilerde kullanılır.

3.1 - Process Seçme

Günümüzde, hemen hemen tüm kurulu sistemlerde, bir istek meydana geldiği anda, işe hizmet verilmesi ve sonuç alınması aşamaları arasında, işin geçmesi gereken çeşitli

aşamalar vardır. Aşağıda bu aşamaların temel olanları açıklanmaktadır (Gordon.B.Davis, 1982);

- .Data-Preparation.: Verinin giriş için hazır hale getirilmesi.
- .Data-Validation..: Verinin doğruluğunun kontrolü. Bu kontrol giriş sırasında yada girişten sonra yapılır.
- .Input.....: Verinin bulunduğu ortamdan ya da terminalden okunması.
- .Processing.....: Verinin düzenlenmesi ve hesaplamalar yapılması.
- .Normal Output....: Çeşitli raporlar çıkarılması ya da bunların belli bazı ortamlarda saklanması.
- .Error Output.....: Meydana gelen hataların belli bazı ortamlarda saklanması. Bunlar Data-Validation, Input ya da Processing sırasında oluşabilir.

Seçilen iş türü ne olursa olsun yukarıda belirtilen aşamalardan geçmek zorundadır. Burada tür olarak iki

iş'ten sözetmek olasıdır; süreli ve anında (bunlara BATCH ve ONLINE işlerde denilebilir). Aşağıdaki listede bu iki tür işde, yukarıda sözü edilen altı fonksiyonun hangi durumlarda olması gerektiği belirtilmiştir(GORDON.B. 1982);

FONKSİYON

ZAMANALAMA SECENEKLERİ

DATA PREPARATION ----- Veri, periyodik işlem için batch olarak saklanır veya anında terminalden girilebilir.

DATA VALIDATION ----- Veri, terminalden giriş sırasında ya da bilgisayar tarafından daha sonra kontrol edilir.

INPUT ----- Giriş verisi, olduğu andaki ya da periyodik bir şekilde işlenen batch veri olabilir.

PROCESSING ----- Programın çalışması, ya her bir hareket olduğu anda ya da batch verilerde periyodik olarak yapılır.

NORMAL OUTPUT ----- Normal çıktı, ya direk yazıcıdan alınır ya ekrana verilir ya da daha sonraki kullanımlar için depolama birimlerine kaydedilir.

ERROR OUTPUT ----- Hata çıktısı, ya anında yazıcıdan alınır ya ekrana verilir ya da daha sonra hata analizinde kullanılmak üzere saklanır.

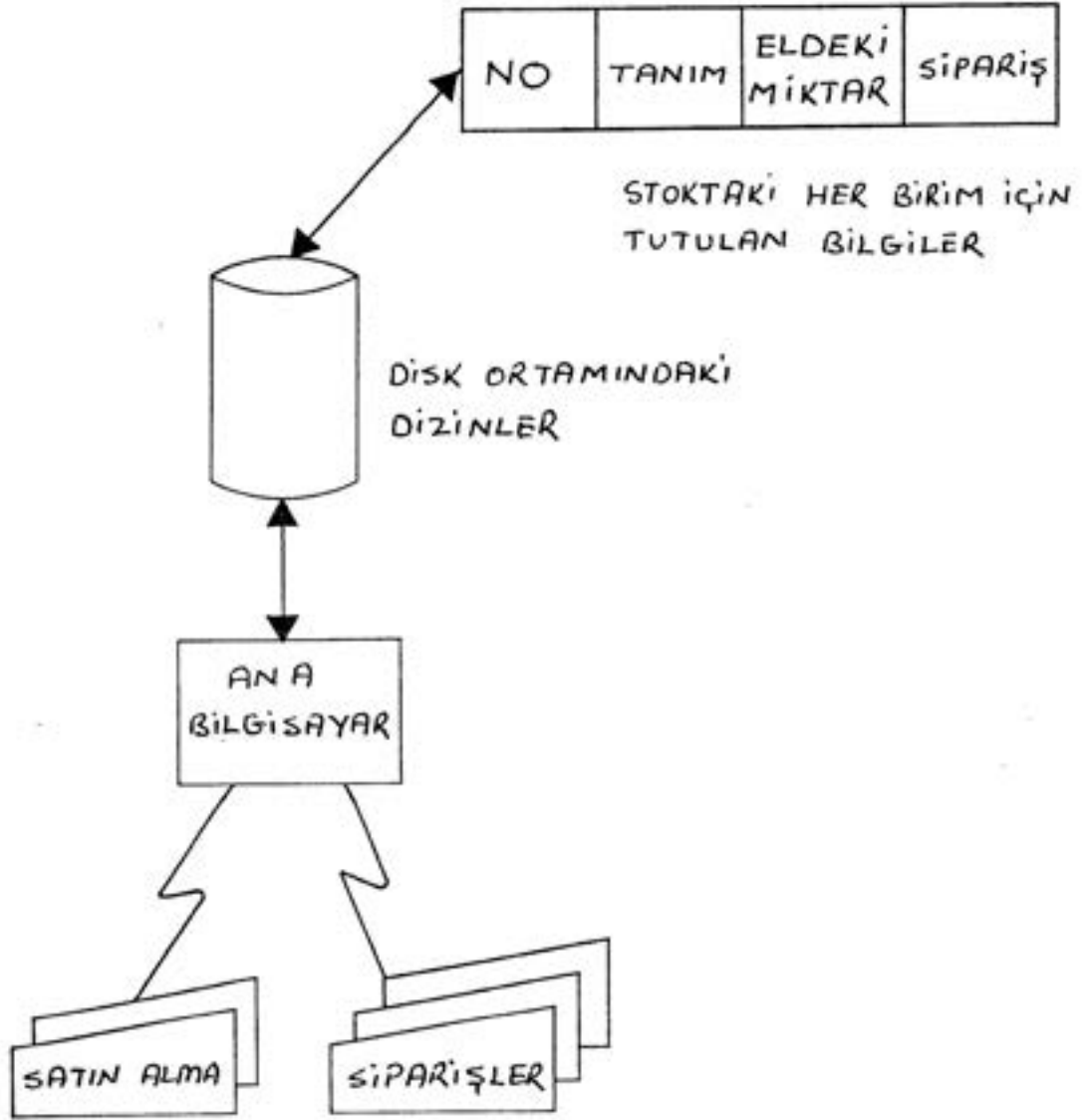
Son zamanlarda üretilen birçok bilgisayar çeşitli amaçlarla kullanılmaktadır. Bu kullanımlardan birisine örnek olarak, Amerikada birçok sistemin veri depolayan bir gereç olarak kullanılmasıdır. Bunun en büyük nedeni, bu işlere ayrılan paranın son yıllarda oldukça artmasıdır. Yapılan araştırmalara göre, kazanılan her sekiz doların bir doları, kayıt tutmakla ilgili olan harcamalar olarak sarf edilmektedir. Bu tip bir sistemde çalışabilecek olan iki tür işleyiş, (Bölüm 3.2 ve 3.3)'de verilmiştir (GORDON.B.Davis,1982).

3.2 - Online Processing

Kurulu bulunan bazı sistemlerde, işin yapısı gereği, halen daha çalışmakta olan batch sistemler vardır. Bu mantık her zaman yapılacak olan işin yapısına uygun düşmez. Örneğin herhangi bir fabrika yönetiminde veya bankacılık uygulamalarında olduğu gibi. Yani yapılacak olan iş, anında yanıtlanması gereken iş olabilir. Bu yapıda herhangi bir anda oluşacak bir isteğe, oluştuğu anda hizmet

verilmelidir. Bu tür çalışma şekline Online-Realtime (OLRT) denir. Bu ortamı sağlayabilmek için, ana bilgisayarımıza bağlı yeteri kadar terminal ve bu terminallerden bilgi girilip gerekli ekleme, silme veya güncleme işlemlerinin yapılabilmesini sağlayacak programlara gereksinim vardır. Örneğin;

Sistemde bulunan ve ilk başvuru ile ilgili olan kişiye bir müracaat olduğu anda, sorgulama veya hareket başlayacaktır. İsteğe bağlı olarak sorgulama ya da herhangi bir işlem olacaktır. Son olarak da, sorgulamalardan ve işlemlerden sonra, elde edilmiş olan veriler ve sonuçlar ilk işlemi başlatan kişiye geri gelecektir. Bu tür bir çalışmada yanıt süresinin kısalığı oldukça önemlidir. Bunu sağlayabilmek için bu gibi sistemlerde tape ortamının yerine disk ortamının bulunması gereklidir. Burada zaman zaman alınması gereken hareket ve veri listelerinin sadece backup (yedekleme) amacı vardır ve geri dönüş için yararlı olacaktır. Yukarıda anlatılan çalışmaya ilişkin bir örnek (Şekil 3.1) 'de verilmiştir (GORDON.B.DAVIS,1982).



OLRT ÇALIŞMA ŞEKLİNE BİR ÖRNEK
ŞEKİL 3.1

3.3 - Batch Processing

Bilgisayarların gelişmeye başladığı ilk zamanlarda, sistemde bir anda sadece bir program çalışır durumdaydı. İkinci bir programın çalışabilmesi için, birinci programın bitmesi ve ikinci program için hazırlıkların elle yapılması gerekiyordu. İkinci kuşak bilgisayarlar ve giriş/çıkış birimlerinin gelişmesiyle bu olay ortadan kalktı. Bu sayede çalışacak işleri kuyruklamak ve makinayı durdurmadan sıradan çalıştırmaya başlamak mümkün oldu. Böyle bir ortam ise ancak şu koşullarda olabilirdi ;

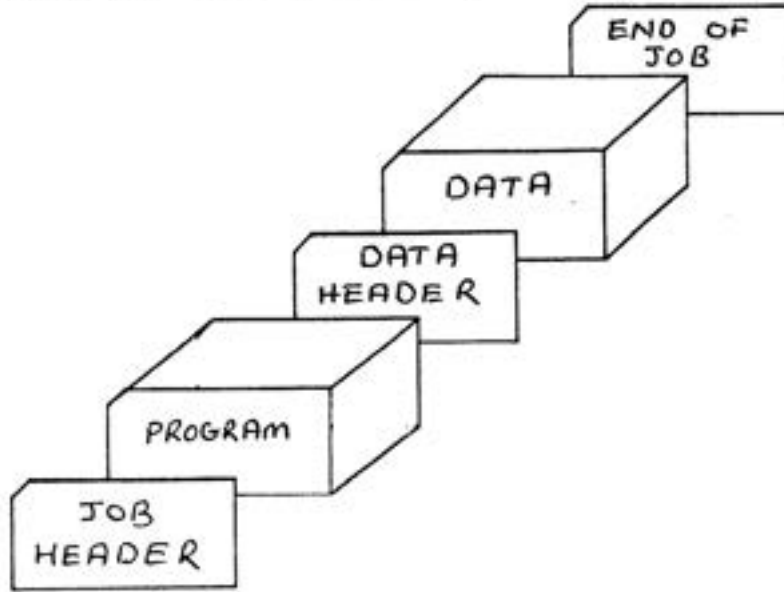
- .Çalışacak işleri bir tape'de tutmak,
- .Makinayı çalıştırmak,
- .Oluşacak sonuçları başka bir tape'de tutmak.

Yukarıda çok kısa bir örneği verilen çalışma şekline Batch Processing denilmektedir.

Bu çalışma şeklinde, ana bellekte daima olması gereken bir Monitor yada Supervisory program vardır. Bu programın ana bellekte çalışma zorunluluğundan ana belleğin kapasitesinin artırılması gerekli olmuştur (JOHN P.HAYES,1978).

Batch processing işlemine verilen yukarıdaki örnek, tape yan birimini kullanmaktadır. Bilgisayar

sistemlerinde giriş birimi olarak kullanılan Okuyucu'da ise olay biraz daha dikkat gerektirmektedir. Bu mantıkta da işler, okuyucuda buldukları sırada işlenmektedir. Bundan dolayı da okuyucuda buldukları format oldukça önemlidir. Bu yapı (Şekil 3.2)'de verilmiştir (V.C.HAMACHER,1978).

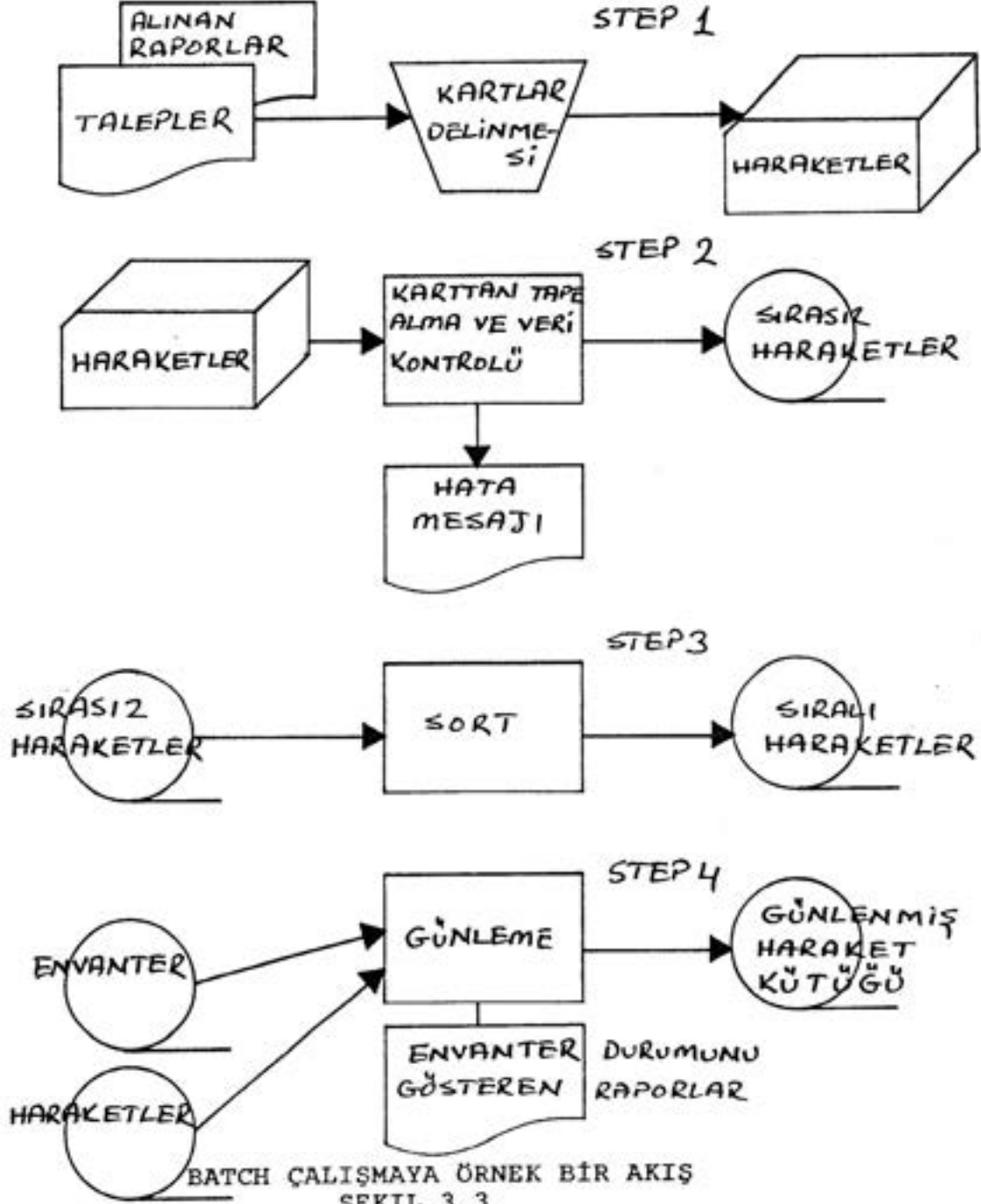


BATCH ÇALIŞMADA OKUYUCU FORMATI
ŞEKİL 3.2

Batch processing mantığında oluşan hareketler, günümüzde de çokça kullanılan tape biriminde sırasız olarak derlenir.

Daha sonra oluşan bu tape kayıtları bir sıraya konur, son olarak da ilgili kütük günlenir. Bu işlemten sonra da hareketlere bağlı olarak meydana gelmiş olan değişiklikler bir liste halinde yazıcıdan çıkarılır. Günlük hayatta

uygulanan yöntemler yukarıda sözü edilenlerden daha karmaşık olmaktadır. Konuyla ilgili basit bir örneğin akış şeması (Şekil 3.3)'de verilmiştir.



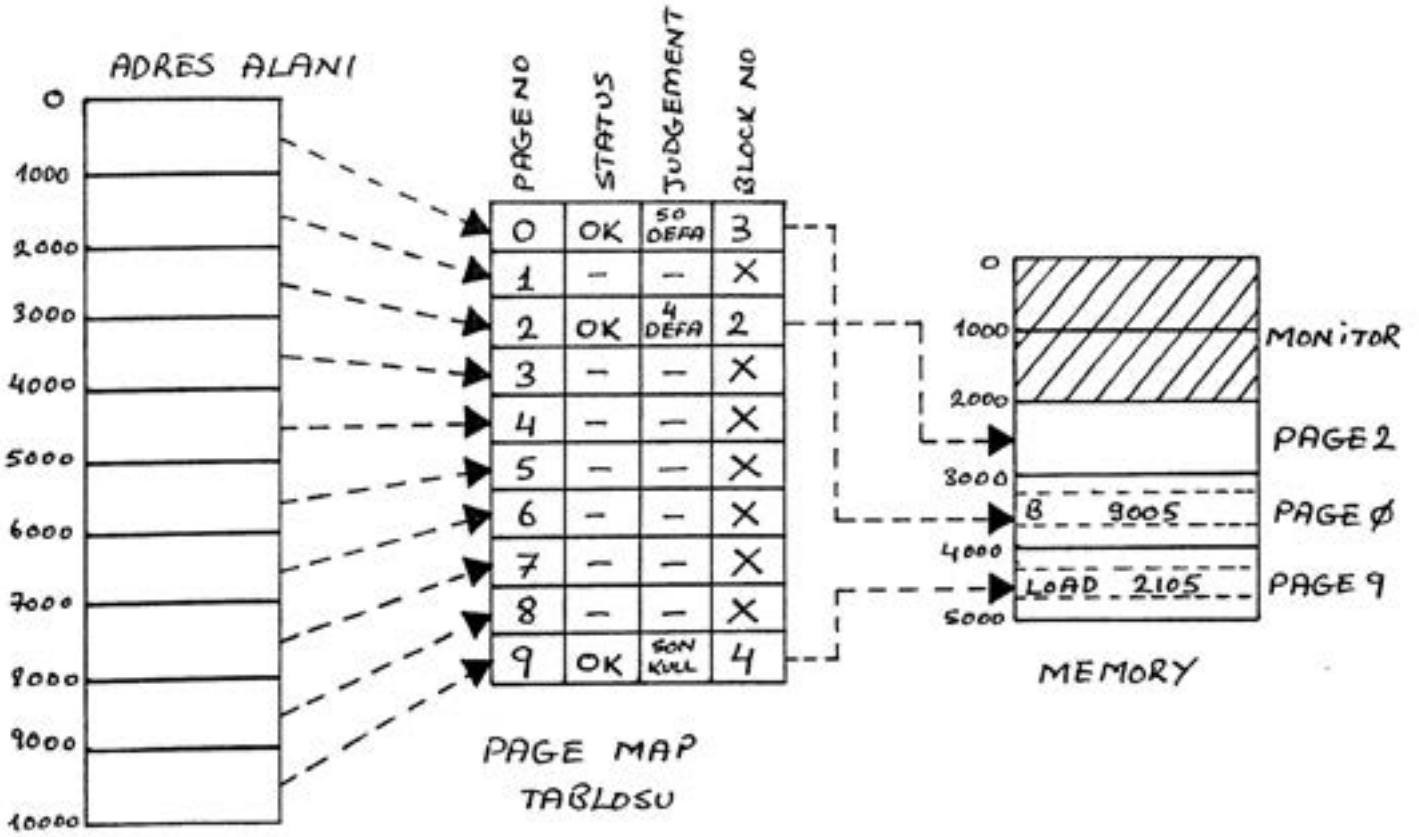
4 - VARSAYILAN BELLEK

Varsayılan bellek kavramının temel özelliği, sistemde var olan ana bellek kapasitesini sınırsız kılmaktır. Bu sistem, ana bellekte bulunan ve çalışmakta olan programların gereksinim duyabilecekleri program bölümlerini, çok hızlı yan birimler olan diskler üzerinde tutup, gerektiğinde bunlara refere etmekden oluşmaktadır.

4 - 1 Varsayılan Bellek Nedir ?

Genelde Main Frame tipi olan bilgisayarlarda kullanılan ve kullanıcıya sonsuz bellek tahsisi yapılmış rahatlığı veren bir donanım şeklidir. Buna DEMAND PAGED ALLOCATION denir. Varsayılan Bellek (VIRTUAL MEMORY), bu donanım altında çalışan programın kullandığı adres alanının bir başka şekilde tanımlanmasıdır.

Elimizde 10000 byte'lık adresleme alanına gereksinimi olan bir program olduğunu varsayalım. Buna karşın, ana bellek kullanılabilir boş alanı 3000 byte olsun. Bu 10000 byte'lık programın, 3000 byte'lık ana bellekte, Demand Paged Allocation mantığına bağlı kalmak kaydıyla çalışması mümkündür. (Şekil 4.1) 'de bu kullanımın nasıl olacağı görülmektedir (JOHN.J.DONOVAN, 1972).



DEMAND PAGE ALLOCATION GÖSTERİMİ
ŞEKİL 4.1

Bu donanım mantığında, işlemin yapısı gereği, fazlaca kullanılan Page Map'ler en önemli birimi oluşturmaktadır. Kullanılan bu tabloların formatları aşağıdaki gibidir.

- Sayfa (page) numarası,
- Block Numarası (ana bellekteki adres),
- Statü (sayfanın bellekte olup olmadığını gösterir),
- Judgement (sayfanın kaç kez kullanıldığını gösterir).

(Şekil 4.1)'de çalışma şeması gösterilmiş olan 10000 byte'lık işin, çalışmasına sıfırdan sayfadan başladığını kabul edelim. Programın akışı sırasında 9005 adresine bir transfere rastlanacaktır. Bu durumda donanım, 9005 adresindeki 9 numarasını index olarak kullanarak sayfa tablosuna ulaşacaktır. Daha sonra yapılacak olan işler sırasıyla aşağıdaki gibi sıralanabilir.

- İstenen sayfanın bellekte olup olmadığı kontrol edilir,
- İstenen sayfa bellekte yoksa, bu durumda sayfa 9 istenir (demand) ve daha önce bellekte bulunan sayfaların herhangi birinin üzerine yerleştirilir. Bu yerleştirme, Judgement biti kontrol edilerek hangi sayfanın yerleştirmeye uygun olduğu bulunarak yapılır. Yani en küçük judgment değeri olan sayfanın üzerine yerleştirme yapılır.

Burada istenen sayfanın STATUS not = OK ise, işletim kesilir ve bir kesinti yaratılır (PROCESSOR INTERRUPT yada PAGE INTERRUPT). Bunun üzerine işletim sistemi (burada MONİTOR olmaktadır) işletimi eline alır ve ikincil belleğe ulaşır istenen sayfayı bulur. Bu sayfanın belleğe gelmesine

kadar geçecek sürede ise işletim sistemi başka program yada işlerle ilgilenir (JOHN.J.DONOVAN, 1972).

4 - 2 VM / SP Hakkında Bilgi

Virtual Machine / System Product, IBM bilgisayarları tarafından desteklenen bir işletim sistemidir. Bir kaynağın birçok kullanıcı tarafından kullanılmasını sağlar. Kullanıcı açısından bakıldığında, sistemin tüm kapasitesini kullanabilme görüntüsü veren bir sistemdir. Burada kelime olarak geçen 'görüntü' sözcüğünden dolayı bu işletim sistemine **Virtual Machine** denmiştir. Daha öncede belirtildiği gibi bu sistem, kullanıcılara sistemde gerçekte bir tane olan bir birimi tek başına kullanıyormuş ve çeşitli işlemler yapabiliyormuş hissi verir.

Virtual Machine kavramı ile sistemde zaman paylaşım ve çoklu programlama olanakları, tüm sistem için yaygınlaştırılmıştır. Sistemde bir anda birden çok CMS makinası çalışabildiği gibi, aynı zamanda DOS işletim sistemi bile çalışabilmektedir. Sözü geçen tüm varsayılan makinalar, sistemde bulunan gerçek makinanın bir kopyasıymış gibi çalışabilir ve onun güçlü işletiminden yararlanabilir.

Sistemde çalışmakta olan işler, altında çalıştıkları işletim sistemi tarafından kontrol edilir. Bu işletim sistemlerinin hepsi VM/SP tarafından izlenir ve gerektiğinde kontrol altına alınır. Her bir işletim sisteminin çalışabilmesi için gerekli olan kaynak ve zaman paylaşımının kontrolünü VM/SP yapar (IBM GENERAL INFORMATION, 1983).

Belirtilen Virtual Machine'ler;

- Değişik konfigürasyonlarda olabilir,
- Değişik işletim sistemi altında çalışabilir,
- Batch yada interactive işlem yapabilir,
- Başka bir VM ile aynı anda çalışabilir.

Buraya kadar sözü edilen konulardan da anlaşılacağı gibi, VM/SP çok çeşitli kullanımlara olanak sağlamaktadır.

VM/SP'de;

- Tipik bir kullanıcı yoktur,
- Organizasyondaki tüm kullanıcılar için olanaklar aynı şekilde sağlanmaktadır,
- Her kullanıcı kendi işine en uygun olan işletim sistemini seçebilmekte ve bunun altında çalışabilmektedir. Kurulu bulunan herhangi bir VM/SP işletim sisteminde çalışabilecek işletim sistemleri şunlardır;

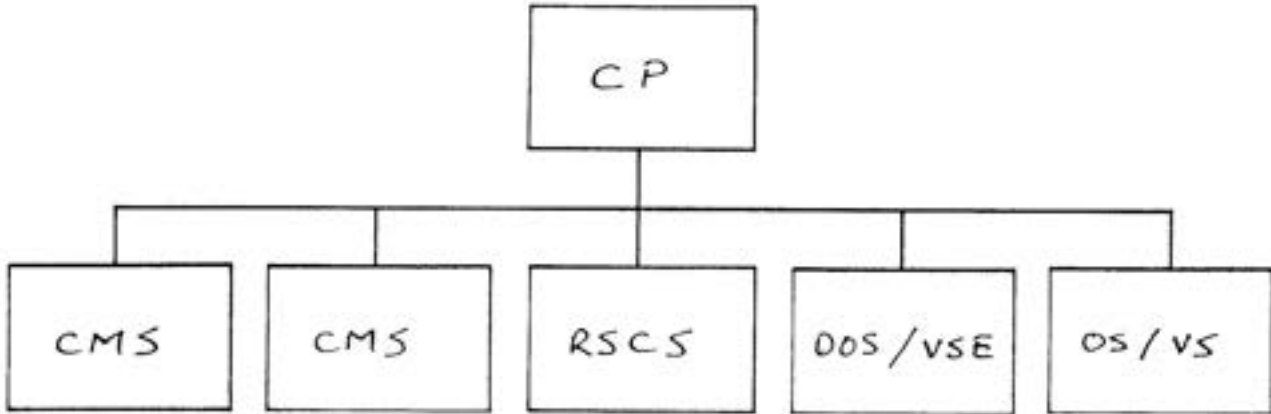
- .DOS, çeşitli release ve versiyonları,
- .OS/VS, çeşitli release ve versiyonları (OS/PCP ile OS/VS2 arasında olanlar),
- .VM/SP işletim sisteminin kendisi,
- .CMS,
- .RSCS

Yukarıda belirtilen konular VM/SP 'nin en temel farklılık ve üstünlükleridir. Bir VM/SP işletim sisteminde bunlardan başka olması gereken en temel özellikler, CP ve CMS programlarıdır. Bu iki yazılım, VM sisteminin daha etkin kullanılmasına yardımcı olmaktadır.

CP (Control Program) VM/SP işletim sisteminin en temel bileşenidir. Bu program, bir kaynak denetim programıdır. Bundan dolayı da bir kaynak kontrol denetleyicisi olarak da adlandırılır.

CMS (Conversational Monitor System) VM/SP işletim sisteminin kullanıcılar açısından en temel bileşenidir. VM/SP sistemini kullanan kullanıcıların sistemle olan ilişkisini kolaylıkla sağlayan denetleyici programdır.

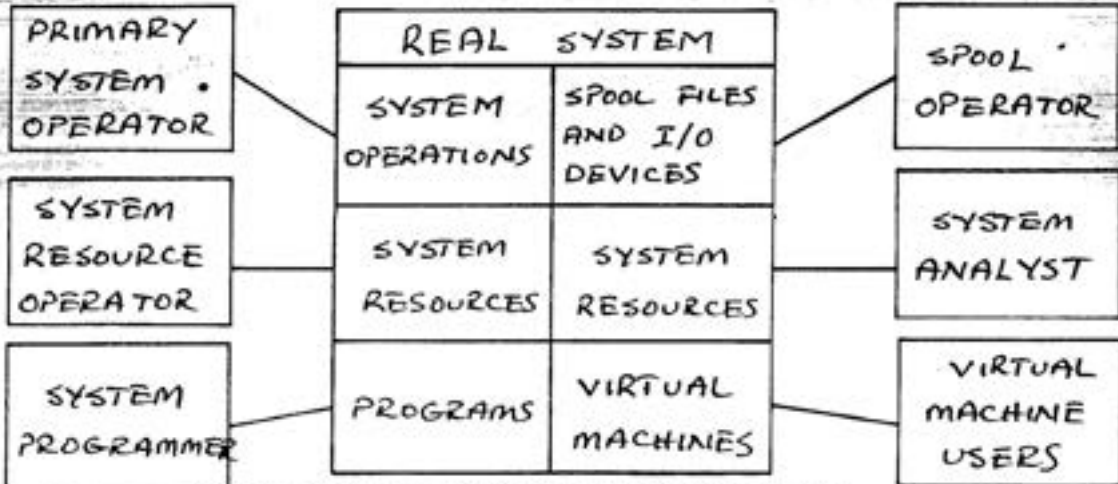
4 - 3 CP Hakkında Bilgi



CP ORTAMINA BİR ÖRNEK
ŞEKİL 4.2

(Şekil 4.2)'de görüldüğü gibi CP, değişik işletim sistemleri altında çalışan VM'lerin, gerçek makinanın kaynaklarından yararlanmalarını sağlamaktadır.

Burada dikkat edilmesi gereken konu, CP'ye komut verirken, CP'nin verilen komutun ayrıcalık derecesini kontrol etmesidir. Ayrıcalık dereceleri (Şekil 4.3)'de gösterilmiştir;



CP ORTAMINDA AYRICALIK DERECELERİ
ŞEKİL 4.3

F özelliğinde olan ayrıcalık derecesi, sistem ile ilgili olan çeşitli verileri kullanabilmekle ilgilidir. Bu özellikte olan VM'leri ancak servis yetkilisi kullanabilmelidir. Diğer özellikler ise, yetki açısından azalan bir sırada A-G arasında belirlenmektedir.

4 - 4 CMS Hakkında Bilgi

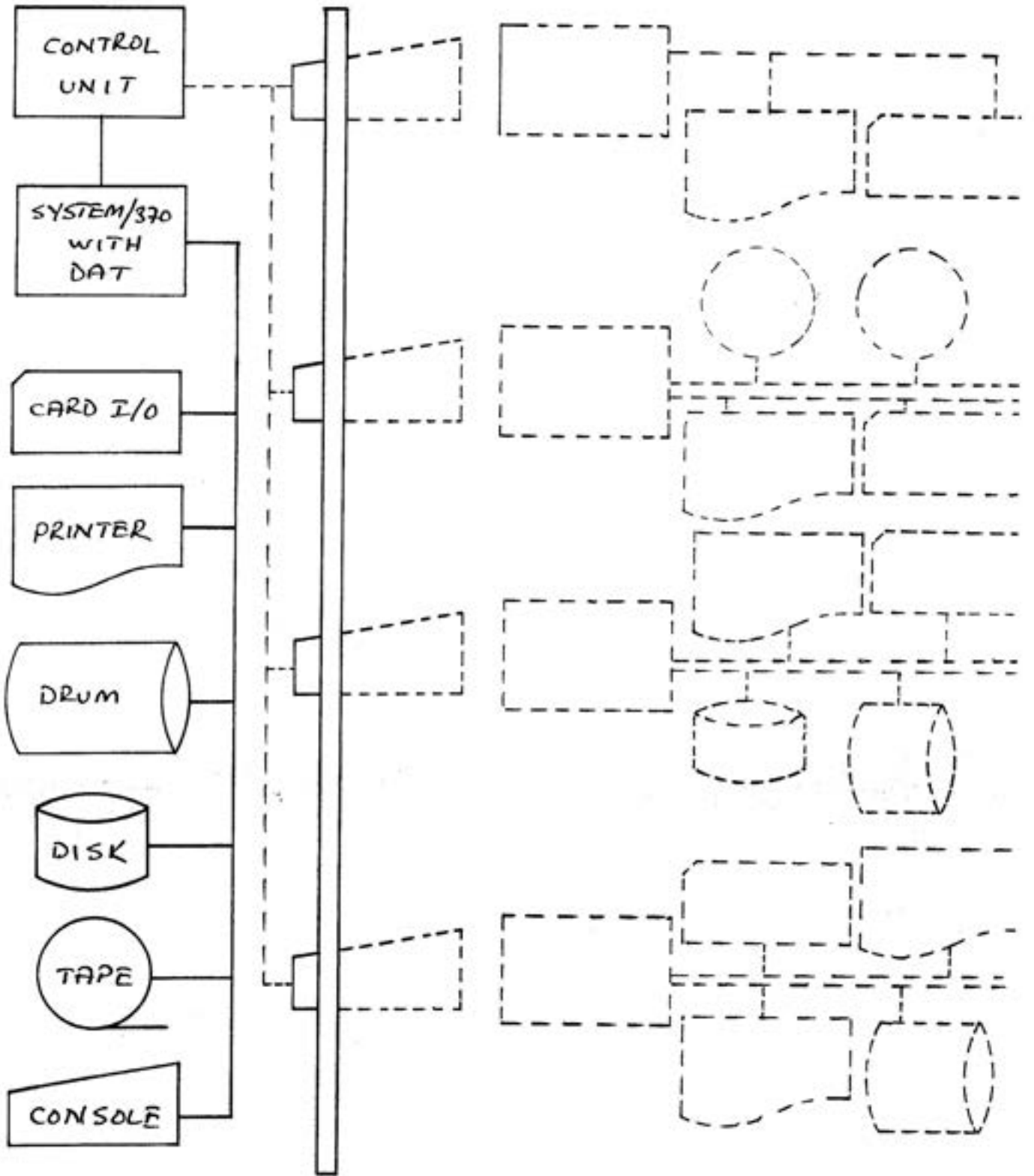
CP'nin tamamlayıcısı şeklinde tanımlanabilecek olan CMS, her bir Virtual Machine'de çalışabilecek, tek kullanıcıyı bir işletim sistemidir. CMS'in VM/SP altında çalışabilmesi için CP programı gerekmektedir. CMS, kullanıcılara OS ve DOS disklerine ulaşımında ve çeşitli işlemler yapmalarında yardımcı olmaktadır. CMS'in, kullanıcılar açısından temel olarak sağladığı yarar, terminal tabanlı karşılıklı iletişimli program çalıştırmalarını ve bunların sonuçlarını terminalden görmeleridir. Herhangibir kullanıcı, o anda kullanmakta olduğu VM'den sisteme bir iş yollayıp, daha sonra aynı terminalden başka bir VM açabilmektedir. CMS sayesinde kullanıcı, çalıştırmak istediği bir işi, başka bir VM'e yollayıp, orada çalışmasını ve sonuçların tekrar kendi VM'ine gelmesini sağlayabilmektedir. Bunun getirdiği yarar

işin çalışması sırasında kullanıcının konsolunu başka işler için kullanabilmesidir. İşlerin yollandığı VM ise CMSBATCH özelliğini desteklemelidir.

Virtual Machine aşağıdaki gibi özetlenebilir;

- İşlevsel olarak gerçek bilgisayarın bir eşidir,
- VM, CP'nin kontrolü altında, gerçek sistemde çalışan yazılım ve simüle edilmiş donanım kaynaklarına sahiptir.
- Her bir VM, VM/SP direktörüsünde tanımlı olmalıdır. VM/SP direktörüsünde; varsayılan bellek miktarı, I/O birimleri ve Console tanımı bulunur. (IBM, GENERAL INFORMATION REL.3, 1983)

(Şekil 4.4)'de yukarıda sözü edilen olayları şematik olarak görmek olasıdır.



VM İŞLETİM SİSTEMİ MANTIĞINA ÖRNEK
ŞEKİL 4.4

5 - CMSBATCH NEDİR ?

CMSBATCH makinası, bulunduğu sistemdeki tüm kullanıcılara açıktır. Sistem operatörü, ilgili makinanın kullanıcı kodunu kullanarak CMSBATCH makinasını devreye sokar. Operatör daha sonra CMSBATCH komutunu kullanarak açmış olduğu VM'i batch mode'a çevirir. Özellikle çok uzun süreli çalışması gereken çeşitli programları çalıştırmak zorunda olan kişilere daha yararlı olabilmektedir. CMSBATCH makinasını kullananlar FIRST-IN-FIRST-OUT mantığına göre işlem görürler. CMSBATCH makinası, her işe başlamadan önce, bir önceki işin kullandığı tüm sahaları silmek için, iki işin arasında CMSBATCH programını yeniden yükler ve kendi kendisini çalıştırır.

5.1 - CMSBATCH Özelliğini Kullanmak

CMSBATCH makinasının temel özelliklerinden bir tanesi, çalışmakta olan işin, işi yollayan kişinin makinasındaki gibi çalışmasıdır. Bir programın CMSBATCH makinasına yollanabilmesi için bazı iş kontrol deyimleri gerekmektedir. Herhangibir sistemde kurulu bulunan CMSBATCH makinasının VM ismi, sistem operatörü tarafından bilinir ve bu kişiden öğrenilebilir. VM/SP işletim sisteminin yapısı gereği, sistemde zahiri okuyucular bulunmaktadır. Bu okuyucular her VM'e zahiri olarak bağlı çalışabildiği

gibi, sistemin de kendi kullanımı için gerçek ve zahiri birimleri bulunabilmektedir.

CMSBATCH makinasına herhangi bir işin yollanması ise, bu varsayılan okuyucular ile olmaktadır. Yani ya sistemin okuyucusunu yada direk olarak CMSBATCH makinasının okuyucusunu kullanarak olmaktadır. Sistemin yada bir VM'in okuyucusunda bir kütük çeşitli formatlarda bulunabilir. Ancak, eğer CMSBATCH makinasına bir iş yollanması isteniyorsa bu durumda bu format aşağıdaki gibi olmalıdır;

- .İş kontrol deyimleri (JOB deyimi),
- .Yollanmak istenen program (çalışması sırasında özel kütükler kullanıyorsa, FILEDEF komutları),
- .Yollanan program (çalışması sırasında işi yollayan kişinin bağlı olduğu disk ünitelerini kullanıyor ise LINK ve ACCESS komutları),
- .Program, çalışması sırasında çeşitli sonuçlar üretiyorsa ve bunların işi yollayan kişiden başka bir kişiye geri yollanması isteniyorsa, SPOOL komutu.

Batch makinasının işleyişini kontrol etmek için yollanan iş akış deyimlerinde bazı CP ve CMS komutları da kullanılabilir. Bu, bir EXEC programı kullanarak sağlanabilir.

Batch makinası bir işin çalıştırılması için başladığı anda, console datasını otomatik olarak bir SPOOL kütüğünde tutmaya başlar. Yollanan programın çalışması bittiğinde eğer herhangi bir çıktı oluştuysa, bu kütük aşağıdaki şekillerde işi yollayan kişiye iletilir;

- .İşi yollayan kişinin kullanıcı koduyla belirlenmiş, biriktirilmiş console çıktısı ve çıktı kütükleri sistem görüntü yazıcısına yollanır,
- .İşi yollayan kişinin kullanıcı koduyla belirlenmiş olan PUNCH kütükleri ise sistemin görüntü CARD PUNCH'ına yollanır.

Eğer CMSBATCH makinasına iş yollanırken SPOOL komutu kullanıldıysa, çalışma bittikten sonra oluşan kütükler bu komutta belirlenen VM'e yollanır. Diyelim ki sonuç kütükleri ve console çıktılarının SYSADM5 makinasına yollanılması isteniyorsa, bu durumda SPOOL komutu aşağıdaki gibi olmalıdır;

CP SPOOL PRT FOR SYSADM5

Sistem operatörü, sistemde bulunan ancak henüz kullanıma geçmemiş olan CMSBATCH işleri için PURGE ve ORDER komutlarını kullanabilir. İş yollayan kişi, ancak yolladığı işle ilgili olmak üzere CLOSE, PURGE, CHANGE veya ORDER komutlarını batch işlerinin kontrolünde kullanabilir.

5.2 - VMBATCH Sistemi Nedir

VMBATCH altsistemi, IBM kullanıcılarının isteğine bağlı olarak sistemlere verilen ve paralı olan bir üründür (IBM INTRODUCTION,1984). Bu ürün, VM/SP'deki CMSBATCH VM'lerini izler, başlatır ve sıraya sokar. Bu sayede aynı anda birden fazla CMSBATCH makinası çalışabilir durumda olabilmektedir. Bu çalışma sırasında VMBATCH altsistemi, sistemdeki işleri sınıflarına bağlı olarak sıraya sokar. Sisteme bir iş yollandığı anda, operatör yada kullanıcı, başlama zamanını ve önceliğini belirler. Sistemde mevcut olan CMSBATCH makinalarından herhangi bir tanesinin çalışması bittiği anda, VMBATCH makinası kuyruktaki mevcut işlerden birtanesini seçer ve boşalmış olan makinaya yollar. CMSBATCH makinalarının çalışması sırasında VMBATCH makinası bunları çeşitli aralıklarla denetler. Bu denetlemeler sırasında ise yollanan işin gerçekten

çalışıp çalışmadığı, işe başlamadan önce kullanıcı tarafından belirlenen maksimum çalışma ve çıktı satırını geçip geçmediği kontrol edilir.

5.2.1 - VMBATCH Altsisteminin Kullanılması

Sistemde VMBATCH makinasına yollanan işler genelde CMS EXEC kütükleridir. Yollanan bu kütükler aynı zamanda, batch makinasının kullandığı minidisk üzerine yerleştirilecek olan veri kütüklerindeki içerebilir.

VMBATCH altsistemi sayesinde, uzak bir sistemden (Remote Job Entry) veya sistem okuyucusundan gelen işler hariç olmak üzere, kullanıcılar iş kontrol deyimlerini hiç kullanmadan işlerini kolay bir şekilde çalıştırabilirler. Uzak bir sistemden veya sistem okuyucusu vasıtasıyla sisteme gelen işlerin ilk kayıtlarında işin sahibini, işin sınıfını, şifreyi ve diğer seçenekleri içeren iş kontrol deyimlerinin olması gerekir. Sistemde, kullanılabilecek basit komutlar yardımıyla kullanıcı, işleri yollamayı, kesmeyi, çalışmakta olan yada çalışan işler hakkında bilgi almayı ve bir işin durumunu öğrenmeyi sağlar.

Kullanıcı, bir iş hakkında bilgi istediği anda, altsistem o iş hakkında o andaki veya son durumunu ve altsistemdeki diğer işlerin sistem kaynaklarının kullanımlarını gösterir.

Bir sistemde VMBATCH altsistemi kullanılarak, sistemin gereksinimleri karşılanabilir. Altsistemin kullanıcılar tarafından yazılmış programların yapmasına izin verdiği bazı işler şunlardır;

- Kullanıcı tarafından tanımlanmış komutları işlemek,
- Bir işin yada komutun işlenmemesi ve bunu ilgili kullanıcıya nedeniyle birlikte rapor etmek,
- Alt sistemi kullanan her kullanıcı hakkında istatistik tutmak,
- Remote Job Entry ile gelen işlerin, geçerli bir noktadan gelip gelmediğini saptamak,
- Hangi kullanıcının, altsistemdeki hangi komutu kullanabileceğini kontrol etmek.

6 - SİSTEM PROGRAMLAMA AÇISINDAN CMSBATCH MAKİNASI

CMSBATCH makinası, VM/SP işletim sisteminin bir programlama ürünü olup, CMS altsisteminde çalışır. CMSBATCH, VM/SP kullanıcılarını, işlerini batch modunda ya kendi okuyucusu yada gerçek sistem okuyucusu üzerinden CMSBATCH makinasına yollayarak çalıştırmalarını sağlar. Daha sonra CMSBATCH makinası, işi yollayan kişiyi, yolladığı işten serbest bırakıp daha başka işlerle uğraşabilmesini veya çalışmasını sağlayıp, yollanan işi başlatır (IBM SYSTEM PROGRAMMERS GUIDE, 1984).

Bir sistemde, RSCS (Remote Spooling Communication Subsystem) ile CMSBATCH makinası aynı anda çalışıyorsa, uzak sistemlerdeki CMSBATCH makinalarını haberleşme hatları üzerinden kullanmak mümkün olabilmektedir. Aynı şekilde CMSBATCH makinasının meydana getireceği sonuçlarda uzak sistemlere yollanabilir.

CMSBATCH makinası, sistemde bu işe ayrılmış olan kullanıcı kodu altında yaratılıp kontrol edilir. Bu sayede yığın işler işlenebilmektedir. Sistem operatörü, CMSBATCH makinasının, IPL sırasında yükleyeceği CMS altsisteminden sonra gireceği CMSBATCH komutu ile batch makinasının çalışır duruma gelmesini sağlar.

CMSBATCH modülü, gerçek yığıt işleyicisi olan DMSBTP TEXT S2 programını yükler. Her bir işten sonra, yani

CMSBATCH makinasına gelen herbir görevden sonra CMSBATCH makinası kendi kendini IPL eder. Bu sayede de sürekli çalışan bir yığıt makinası haline gelmiş olur. Yığıt makinası bu işlemi, CP IPL komutunun PARM seçeneğini kullanarak yapmaktadır. Bu komutu izleyen karakter dizininin, CMS IPL 'ini yapmakta olan CMSBATCH makinasına ait olduğunu anlar. İşler, CMSBATCH makinasının varsayılan okuyucusuna direk olarak kullanıcının terminalinden yollanır ve bunlar geldikleri sırada işlenir.

CMSBATCH makinasının en önemli özelliği, zaman kaybettirici olan işleri, kullanıcının bu makinaya yollayarak interaktif olarak çalışmasına devam edebilmesini sağlayabilmesidir.

Sistem programcısı, CMSBATCH makinasının kullanıcıya daha yakın ve kullanılmasının daha kolay olmasını sağlamak için, bu makinanın çeşitli parametreleri ile oynayabilir yada EXEC programlar yazabilir (IBM SYSTEM PROGRAMMERS GUIDE, 1984)

6 - 1 CMSBATCH Makinasının Tanımlanması

VM sistemlerinde her bir kullanıcının, sistemdeki kaynakları kullanabilmesi ve sisteme iş yollayabilmesi için, sistem direktörisi denilen sahada bazı tanımlamalar yapılması gerekmektedir. Bu tanımlamalar CMSBATCH

makinası için de zorunludur (IBM SYSTEM PROGRAMMERS GUIDE,1984).

Bu tip bir tanımlama aşağıda verilmiştir;

```
USER CMSBATCH BATCH 1M 2M BG
ACCOUNT 13 SYSTEM
OPTION ACCT
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 190 190 RR
MDISK 195 3375 *** 010 'PASSWORD' W 'READPSS'
                        'WRITEPASS' 'ALLPASS'
```

VM sistemlerinde her bir varsayılan makinanın, kendisine bağlı olarak çalışacak olan bir disk birimi vardır ve bu birim varsayılan adresi hep 191 olarak tanımlanmaktadır. Ancak yukarıda da görüldüğü gibi CMSBATCH makinasının 191 disk bulunmamaktadır. Bunun nedenleri ileriki bölümlerde verilecektir.

Bir VM sisteminde, sistemin açılması sırasında kullanıma hazır hale gelmesi istenilen makinalar varsa, bunların hepsini yine sistemin bir başka varsayılan makinası olan AUTOLOG varsayılan makinasının PROFILE EXEC programında bazı tanımlamalar yapılmalıdır. Bu tip bir tanımlamanın CMSBATCH makinası için yapılması istenirse aşağıdaki gibi olmalıdır;

```
AUTOLOG CMSBATCH BATCH CMSBATCH
```

Yukarıdaki tanımlamanın yapılmamış olduğu sistemlerde, sistemin her çalışmaya başlamasından sonra operatörün özel olarak CMSBATCH makinasını açması ve CMSBATCH komutunu girdikten sonra, DISCONNECT etmesi gerekecektir. Bu durumda CMSBATCH makinası çalışır, ancak herhangi terminali işgal etmiyor durumda olacaktır.

6 - 2 CMSBATCH Makinasının Sistem Limitlerinin Değiştirilmesi

VM sistemlerinde tüm makinalar için olduğu gibi, CMSBATCH makinası için de önceden tanımlı olan, ancak değiştirilmesi de mümkün olan bazı limitler vardır. Herhangibir değişiklik yapılmadığı takdirde, CMSBATCH makinası için bu limitler aşağıdaki gibidir (IBM SYSTEM PROGRAMMERS GUIDE,1984)

```
Varsayılan İşleyici zamanı...: 32767
Punch Kart çıktısı.....: 32767
Satır Yazıcı çıktısı.....: 32767
```

Yukarıdaki limitlerin herhangi bir sebeple değiştirilmesi gerektiği zaman BATLIMIT MACRO kütüğü kullanılmalıdır. Bu kütük CMSLIB macro kütüphanesinde bulunmaktadır. Herhangibir değişiklik yapıldığında, yapılmış olan değişikliklerin etkili hale gelebilmesi için DMSBTP programının yeniden derlenmesi gerekecektir.

6 - 3 CMSBATCH Makinasına Bazı Özel Durumları Kontrol Etmek için Altyordamlar Yazılması

CMSBATCH makinası, kullanıcının tanımladığı ve kullandığı kontrol dillerini ve bunun yanında /JOB kontrol kartlarını detekleyebilir. Bu destekleme mekanizması, kullanıcının faydalanabileceği dışsal altyordamlar sayesinde olmaktadır. Bunun kullanılabilmesi için, sistem programcısının gerekli iki altyordamı geliştirmesi gerekir. Yazılması gereken bu altyordamların isimleri BATEXIT1 ve BATEXIT2, tipleri TEXT ve eğer sistem diski üzerindeyse mod numarasının 2 olması gerekir. Sistem programcısı tarafından yazılan bu programlar ASSEMBLER dilinde yazılmış olmalı ve 12'inci yazmaç dahil tüm yazmaçları saklamalıdır. Bu sayede CMSBATCH makinasının adreslemesinde saklanmış olur. Bu altyordamlar sistem diski üzerinde ise, CMSBATCH makinasının her iş bitiminde yeniden yüklenmesi sırasında tekrar yüklenir (IBM SYSTEM PROGRAMMERS GUIDE,1984).

6.3.1 - BATEXIT1 Altyordamı

Kullanıcılar tarafından kullanılmış CMS kontrol deyimlerinin dışında herhangi bir deyim olmamasını kontrol eden bir altyordamdır. Örneğin, bir FORTRAN programını derleyip, link ettikten sonra çalıştıracak OS iş kontrol dilini tarayacak bir altyordam yazılabilir.

BATEXIT1, CMSBATCH makinasının okuyucusuna gelen işlerin her okunuşundan sonra devreye girer. Genel amaçlı birinci yazmaç CMSBATCH makinasının varsayılan okuyucusunun adresini içerir. Burada da batch makinası tarafından işlenecek işin ilk kartı bulunur. Bu sayede de BATEXIT1 altyordamının her bir kartı teker teker kontrol etmesi sağlanmış olur. Eğer okuyucudan okunan kartta herhangi bir hata söz konusu ise, BATEXIT1 altyordamından geri gelen sonuç 15 inci yazmaçta bulunur ve 0'dan farklı bir geri dönüş kodunu içerir. Bu durumda CMSBATCH hatalı olan bu kartı ihmal edip bir sonraki kartı okur. BATEXIT1 altyordamından geçen kartın 15 inci yazmaçtaki geri dönüş kodu 0 ise bu durumda CMSBATCH bu kartı CMS'e, işlemek üzere geçirir (IBM SYSTEM PROGRAMMERS GUIDE,1984).

6.3.2 - BATEXIT2 Altyordamı

CMSBATCH makinasına yollanan iş kontrol kartlarındaki /JOB kartının ek bilgilerle kullanılmasını sağlamak için yazılmıştır. BATEXIT2 altyordamı, CMS'in gerekli parametre listesini oluşturmasından sonra, VM/SP'nin /JOB kartının gerekli işlemlerini ve yiğit özelliklerini işlemesinden önce, kontrolü eline alır. Kontrol BATEXIT2 altyordamına geçtiği anda 1 numaralı yazmaç, CMS parametre

listesi tamponunun adresini içerir. Bu liste, /JOB kartındaki her bir eleman için 8 byte'lık seri tampondan oluşmaktadır. Okuyucudan okunan kartın işlenmesi sırasında bir hata meydana gelirse, 15 numaralı yazmacın içeriği sıfırdan farklı olur ve ilgili iş kesilir. Eğer 15 numaralı yazmacın içeriği sıfır ise bu durumda kullanıcının kullanım kodu doğru demektir ve ilgili iş çalışmaya başlar (IBM SYSTEM PROGRAMMERS GUIDE,1984).

6.4 - CMSBATCH Makinasında Veri Güvenliği

CMSBATCH makinası, her kullanımdan sonra daha önceden sistemi kullanmış olan kullanıcının tüm verilerini ve kütüklerini silmek için kendisini yeniden IPL (Initial Program Load) yapar. Bu şekilde daha önceki çalışmadan kalan tüm bellek ve kullanım sahaları yok edilmiş olur. Ayrıca önceki çalışmadan bağlanmış olan tüm disklerden de ayrılmış olur. Sistemin tanımlanmasında, CMSBATCH makinasının kendisine bağlı olan herhangi bir disk varsa, bu durumda bu diskler de her işin başlamasından önce formatlanır. Bu sayede de disklerde kalmış olabilecek bilgilerde yok edilmiş ve dolayısıyla önceki kullanıcının verilerine erişmek gibi bir tehlike ortadan kaldırılmış olur. Böylece CMSBATCH makinası için PROFILE EXEC programının çalışması engellenir. Ancak

mutlaka kullanılması gerekiyorsa, bu durumda sistem diski üzerinde bulunan BATPROF EXEC normal PROFILE EXEC için kullanılır (IBM SYSTEM PROGRAMMERS GUIDE,1984).

6.5 - Saklanmış Sistem Kullanmakla IPL Performansının Artırılması

CMSBATCH makinası, her kullanımdan önce kendisini IPL yaptığı için, bu yükü ortadan kaldırmayı kuruluş sırasında düşünerek daha etkili ve hızlı bir IPL işlemi CMS saklanmış sistemi kullanılarak sağlanabilir. Bu da, "sysname" operandında ilgili makinanın ismi verilerek CMSBATCH makinasına geçirilmesi ile sağlanır (IBM SYSTEM PROGRAMMERS GUIDE,1984).

7 - CMSBATCH MAKİNASININ OPERATÖR İLE OLAN İLİŞKİSİ

Bir kuruluştaki olması muhtemel olan bir CMSBATCH makinası, genel olarak sistem odasından açılıp, sistem operatörü tarafından kontrol edilmelidir. CMSBATCH makinası bir kez açıldıktan sonra herhangi bir kontrole gerek göstermeksizin çalışmaya başlar. Operatörün, sadece varsayılan makineyi DISCONNECT durumuna getirmesi yeterlidir (IBM, Operators Guide 1984).

- **Batch Makinasının Başlatılması**
Sistem operatörü CMSBATCH makinasını ilgili ismiyle açtıktan sonra IPL yapabilir.
- **Batch Kullanıcı İsmi**
Kuruluştaki olması muhtemel bir batch makinasının ismi de hem tüm kullanıcılar tarafından kolayca anlaşılır hem de kullanışlı olmalıdır. Bu şekilde kullanıcılar gerekli gördükleri takdirde batch makinasını kullanırlar.

CMSBATCH makinasının açılması ve kullanılır hale getirilmesi sadece sistem programcısının sorumluluğunda olmalıdır. CMSBATCH makinasının sağlıklı çalışması için gerekli olan disklerden varsayılan numarası 195 olan disk mutlaka mevcut olmalıdır. CMSBATCH makinası bu diski her kullanımdan önce formatlayıp bir sonraki iş için boş hale getirmektedir.

7.1 - BATCH MAKİNASININ ÇALIŞTIRILMASI

CMSBATCH makinasının çalıştırılması iki şekilde olmaktadır;

- Sistemin tüm kullanıcılarının bulunduğu kütükteki CMSBATCH ile ilgili olan tanımlama satırına PARM BATCH deyiminin eklenmesi,
- Varsayılan makinanın açılmasından sonra CMSBATCH komutunun girilmesi.

İkinci durumdaki açılışa bir örnek göstermek gerekirse;

```
IPL CMS
CMS mm/dd/yy WED 17.00.00
```

```
CMSBATCH
Y/S (19E) R/O
THE FOLLOWING NAMES ARE UNDEFINED:
    BATEXIT1 BATEXIT2
R;T=0.14/0.39 08:47:40
WAITING FOR THE READER
```

Bu aşamadan sonra operatör isterse makineyi çalışır durumda disconnect edebilir. Batch makinasının özelliğinden dolayı her bir çalışmadan önce kendi kendini IPL eder. Çalışma sırasında meydana gelen tüm hatalar ve kullanılan tüm deyimler, sistemin varsayılan okuyucusuna yollanır ve sistem kapanıncaya kadar burada biriktirilir. İstendiği taktirde bunlar yazdırılarak meydana gelmiş olan bir hata varsa buradan kolaylıkla bulunabilir. CMSBATCH makinesi disconnect

durumda çalışıyor ise, bu kütükte sadece CP komutları gözüktür. CMSBATCH makinasının saklanmış bir sistem olarak çalışması isteniyorsa, bu durumda operatörün varsayılan makinayı açmasından sonra bunu CMSBATCH komutunda belirtmelidir.

7.2 - BATCH MAKİNASININ KONTROL EDİLMESİ

Batch makinasının çalışması sırasında kontrol edilmesini sağlayan bazı komutlar bulunmaktadır. Ancak aşağıda verilmiş olan komutlardan bazılarının da duruma göre kullanım kısıtlamaları vardır.

- CHANGE: Varsayılan okuyucuyu değiştiremeyebilir.
- CLOSE : Varsayılan okuyucuyu değiştiremeyebilir.
- DETACH: Varsayılan havuzlama birimlerini değiştirmede, IPL yada 195 diskinde kullanılamaz.
- DUMP
- DISPLAY
- LINK: CMSBATCH makinası bir anda sadece 10 link kurabilir durumdadır.
Link;
CP LINK userid vaddr vaddr mode password şeklinde kullanılabilir.
- MSG
- QUERY

- SET: Sadece aşağıda belirtilen set komutları kullanılabilir,
ABBREV DOS IMPEX LDRTBLS NONSHARE RDYMSG
SYSNAME UPSI.
- SPOOL: Varsayılan okuyucuyu değiştirmek için kullanılamaz.
- STORE
- TAG

CP komutları, CMSBATCH makinasını kontrol etmede kullanıldığında, CMS kelimesi ile birlikte ve CP'den önce kullanılmalıdır. Bunun nedeni ise, CP komutlarının, makinanın her çalışmaya başlamasında işlem görmesini engellemektir. Aşağıdaki CMS komutları batch makinasının console'unda geçersizdir,

- READCARD
- DISK LOAD
- FILEDEF
- ASSGN

CMSBATCH makinasının çalışması sırasında meydana getirmiş olduğu yazıcı çıktıları, doğrudan sistem yazıcısının varsayılan tampon sahasına yollanır. Sonuç olarak yazıcıdan alınan çıktının formatı şu şekilde olmaktadır.

- CMSBATCH makinasına işi yollayan kullanıcı ismi, kullanıcı ayırıcı kodu
- kütük ismi CMSBATCH

- kütük tipi, eğer /JOB kartında başka isim belirtilmemişse JOB olarak yazılır.

Çalışma sırasında console'a gelen tüm mesajlar ve hatalar biriktirilir ve işin bitiminden sonra yazıcıya yollanır. Bu kütüğün ismi CMSBATCH, tipi ise CONSOLE olur. Ancak havuzlanmış kütük ayrıştırmada veya sonuçların başka kullanıcı ya da uzak terminallere gönderilmesinin istendiği durumlarda CP TAG komutu kullanılmış ise, CMSBATCH makinası bir sonraki işe başlamadan önce tüm havuzlama birimlerini siler.

7.3 - BATCH İŞLERİN BAŞLATILMASI, YENİDEN SIRALANMASI ve BAŞLATILMASI

Batch operatörü, gerektiği zaman sistemin akışını işleri sıralayarak, silerek yada yeniden başlatarak düzenleyebilir. Aynı şekilde bitmiş veya yazıcıya yollanmış işleri, ilgili kişiler sistem yazıcısından yazdırmadan önce silebilir, düzenleyebilir, yada değiştirebilir. Batch makinasında herhangi bir işi silebilmek için aşağıdaki işlemleri takip etmek yeterlidir;

- Bir kez ENTER tuşuna basarak VM ortamına geçin,
- HX komutunu girin,
- Varsayılan makinaı çalışır durumda kapatın.

Yukarıda girilen HX komutu normal olarak çalışmakta olan batch makinasının çalışmasını engeller. Bu şekilde hem kullanıcıya bir hata mesajı yollanmış hemde meydana gelmiş olan anormal durumu belirten bir dump alınmış olur. Bu işlemden sonra batch makinası tekrar kendi kendini yükler ve yeniden çalışmaya başlar. Henüz çalışmaya başlamamış bir işin silinmesi için PURGE komutu gereklidir. Bu komutun formatı aşağıdaki gibidir;

```
PURGE READER spoolid  
      (burada spoolid silinecek olan işin,  
      varsayılan okuyucudaki sıra numarasıdır)
```

Batch makinasının okuyucusundaki işleri istenen bir kritere göre sıraya sokmak için ORDER komutu kullanılabilir. Bu komutun formatı şöyledir;

```
ORDER READER spoolid spoolid...
```

Batch makinasının okuyucusundaki işleri görebilmek için QUERY komutu kullanılır. Bu komutun formatı aşağıdaki gibidir;

```
QUERY READER ALL.
```

Yukarıdaki komut sayesinde operatör, okuyucuda mevcut olan işleri görebilir ve gerekli olan düzeltme, silme yada sıralama işlemini yapabilir.

7.4 - BATCH MAKINASININ DURDURULMASI

Sistemde çalışan batch makinasının çalışmasının normal bir şekilde bitmesinin ve bir başka işe başlamadan durmasını isteniyorsa, sadece HB komutunun girilmesi yeterlidir. Bu komut, batch makinasının çalışmasının normal bir şekilde bitmesini sağlar. Herhangibir nedenle batch makinasının ani olarak kapanmasını ve aynı zamanda işlenmekte olan işin bozulması istenmiyorsa, bu durumda CP SPOOL HOLD komutunun kullanılması gerekmektedir. Bu komutun girilmesinden sonra CP LOGOFF yazılmasıyla batch makinasının çalışması kesilmiş olur. Burada kullanılan HOLD komutu, işlenmekte olan işin tekrar varsayılan okuyucuya yollanmasını sağlar. Sistemde meydana gelebilecek olan herhangi bir aksaklık yüzünden, okuyucuda mevcut olan işlerin bir yedeklemesinin yapılmış olması yararlıdır.

8 - KULLANICILAR AÇISINDAN BATCH MAKİNASI

Herhangibir sistemde bulunabilecek olan CMSEATCH makinası, kullanıcıların çeşitli yığıt işlem yapma isteklerine yanıt verebilecek durumdadır. CMSEATCH makinasının kullanım nedenleri aşağıdakiler olabilir;

- Derlenmesi gereken yada çalışması çok uzun zaman alabilecek olan bir iş varken aynı zamanda terminalde başka bir işle uğraşmak isteniyorsa,
- Herhangibir sebepten dolayı terminalin başında oturmak istenmiyorsa.

8.1 - BATCH Makinasına İş Yollamak

OS yada DCS sistemlerinde, batch mode olarak yollanan tüm işler, JCL(Job Control Language) 'ler ile kontrol edilmektedir. Aynı mantık CMSEATCH makinası için de geçerlidir. Ancak burada JCL kartları yerine /JOB, /SET ve /* kontrol kartlarıyla birlikte CMS komutları kullanılır. CMSEATCH makinası, altında çalıştığı VM/SP sisteminin desteklediği tüm programlama dillerini destekler ve bunların derleme ile çalışmalarını sağlar. Bununla birlikte daha önce sözedildiği gibi bazı CP ve CMS komutlarında kısıtlamalar vardır.

8.2 - BATCH Makinasına Yollanacak İşin Özellikleri

Batch makinasına yollanan işin düzgün çalışabilmesi için gerekli bazı kartların olduğu belirtilmişti. Batch makinasına yollanan ilk deyimin özellikleri şunlar olmalıdır;

- Kart formatında olmalı ve gerçek kartlar kullanarak da girişi yapılabilmeli,
- CMS kütüğü şeklinde olmalı,
- 80 karakter uzunluğunda olmalıdır.

Yukarıdaki formatta hazırlanmış olan bir iş, batch makinasına ya gerçek punch kartının gerçek sistem okuyucusuna okunmasıyla ya da batch makinasının okuyucusuna havuzlanarak yollanabilir. Yollanan bir işin ilk kaydı mutlaka CP ID kartı olmalıdır. Bu kartın formatı şu şekildedir;

ID userid

ID deyimi kaydın ilk karakterinden başlamalı, bunu takip eden (userid) ise sistemde batch makinasının kullanıcı kodunu belirtmelidir. ID deyimi ile (userid) arasında bir veya daha çok boşluk olmalıdır. Örneğin herhangi bir sistemde batch makinasının kullanıcı kodu BATCH1 olsun, bu durumda ID kartının şekli şöyle olacaktır;

ID BATCH1

Hazırlanan işin yolanabilmesi için, sistemde tanımlı olan batch makinasına havuzlanması gerekmektedir. Yukarıdaki örnek batch kullanıcı kodu için kabul edilirse, komut şu şekilde olmalıdır;

```
CP SPOOL PUNCH TO BATCH1
```

8.3 - BATCH Makinasına Varsayılan Kart Girişi

Yukarıda CMSBATCH makinasına nasıl iş yollanması gerektiğinden söz edilmişti. Bu yollardan bir tanesi de CMS PUNCH komutunu kullanmaktır. Bu komutun formatı şu şekildedir;

```
PUNCH BATCH JCL (noheader
```

Yukarıdaki komutun (noheader) seçeneğinin mutlaka bulunması gereklidir. Komutun kullanımı sırasında bu seçenek bulunmuyorsa, yollanan kütüğün ilk kaydının özel bir kayıt olmasının istendiği anlaşılır ve batch makinası bu kayıdı işleyemez. Herhangibir şekilde (noheader) seçeneği kullanılmamış bir kütük gelirse, batch makinası bu kütüğü işleme koymaz ve bir sonraki kütüğü işlemeye devam eder.

Kullanıcı sistemde bulunan yorumlayıcılardan herhangi birini kullanarak istediği işi batch makinasına yollayabilir. Bu şekilde yollanmak istenirse kullanıcının yollamak istediği tüm kütüğün bitimine kadar EXECIO komutu ile satır satır kütüğü yollamalı ve işin sonunda mutlaka CP CLOSE komutunu kullanmalıdır. Yukarıda söz edilen PUNCH komutu kullanıldığı için bununla ilgili varsayılan punch biriminin kapatılması gerekmektedir. ilgili komut ise şu şekildedir;

```
CP CLOSE PUNCH
```

Kullanıcı yorumlayıcı kullanarak işi yollamak isterse, yazması gereken EXEC komutları şu şekilde sıralanmalıdır;

```
CP SPOOL PUNCH CONT  
EXECIO 1 PUNCH '( STRING' /JOB SYSADM5 666666  
PUNCH BATCH JCL '* (' NOHEADER  
CP SPOOL PUNCH NOCONT  
CP CLOSE PUNCH
```

/JOB ve /* KARTLARI

/JOB kartı, batch makinasında çalışması istenen her işin başına konulmalıdır. Bu kart işi yollayan kişinin kullanıcı kodunu ve bununla birlikte işi yapan kişinin genel bilgilerini içermektedir. Bu kartın genel formatı şu şekildedir;

```
/JOB userid acctnum [jobname] [comments]
```

userid: Yollanmak istenen makinanın kullanıcı kodu yada hangi kullanıcı kodu altında yollanmak istenirse onun belirtildiği parametredir. Bu parametre şunları kontrol eder;

- İşin çalışması sırasında sistemin kaynaklarını ve CP zamanının kullanımının hangi kullanıcı kodu altında kayıt edileceğini gösterir,
- Batch işlemin sonunda meydana gelen herhangi bir çıktı varsa bunun yazıcıdan çıkışı sırasında separatörde çıkacak olan isimdir,
- BATCH makinasının çalışması sırasında meydana gelecek olan hata veya her türlü mesajların yollanacağı kullanıcı kodunu gösterir.

Yukarıda belirtilen üç sık, bu kullanıcı için sistem direktörüsünde bir tanımlama varsa geçerlidir.

acctnum: Burada belirtilen numara, her kullanıcı için geçerli olup, sistemin kaynaklarının kullanımını hesap çıkarılan numaradır. Bu numara yollanan işin yarattığı çıktının başında yer alır.

jobname: Bu parametre kullanıcının isteğine bağlı olarak kullanılabilir. O anda çalışmakta olan işi tarif etmektedir. Batch makinasında çalışmakta olan işin meydana getirmesi muhtemel bir çıktı kütüğünün tipini belirler. Meydana gelen çıktının adı daima CMSBATCH olmaktadır.

comments: İstlenen herhangi bir açıklama satırıdır.

Hazırlanan iş kontrol kartlarının sonunu belirtmek için /* karakterleri kullanılır. Batch makinası ilk /*

karakterlerinden sonra gelecek olan ikinci bir /* karakterine kadar olan tüm kartları boş kart olarak değerlendirir. Bu nedenle bir önceki işin /* kartının olmayacağını düşünüp her işin başına /* karakterlerini koymak daha akılcıdır. Aynı kart, herhangi bir diziden bir okuma yaparken kütük sonu karakteri olarak da kullanılmaktadır. Bu sistem, okuyucudan bir kütük transferi yapmada özel bir tekniktir.

/SET Kartı

Bu kart batch sisteminde, kartı kullanan iş için çalışma zamanını, çıktı satır sayısını ve punch satır sayısının toplamını sınırlar. Bu kart aşağıdaki formattadır;

/SET [TIME seconds] [PRINT lines] [PUNCH cards]

seconds: Bir işin varsayılan CPU zamanı olarak ne kadar sürebileceğini gösterir. Bu zamanı geçen iş kesilir.

lines : Bir işin çalışması sırasında oluşturabileceği maksimum satır sayısını gösterir.

cards : Bir işin çalışması sırasında oluşturabileceği maksimum punch kart sayısını gösterir.

Yukarıda sözü edilen değişkenler, herhangi bir tanımlama

yapılmadıđı durumda, her üçü içinde 32767 sabit deđerini kabul eder. Herhangibir tanesinin deđiştirilmesi istendiđinde verilmesi gereken rakam mutlaka bu deđerden küçük olmalıdır. Sistemin çalıřmaya başlamasından önce bu deđerler set edilmelidir. Ancak bunları deđiřtirmeden önce set edilmiř durumlarını bilmek yararlı olabilir. /SET kartı, /JOB kartından sonra iř kartlarının herhangi bir satırında kullanılabilir. /SET kartında kullanılan limitler sadece bu kartın devamında gelecek olan iř için geçerlidir. /SET kartı, toplam üç defa herbir eleman için kullanılabilir. Yani bir eleman için birden fazla /SET kartı kullanılamaz. /SET kartı sadece varsayılan deđerinin deđiřtirileceđi elemanlar için kullanılmalıdır. Bir operand için kullanılan /SET kartı sadece o eleman için geçerli olup diđerlerini etkilememektedir.

9 - BATCH MAKİNASININ ÇALIŞMASI

CMSBATCH makinası çalışmaya başladıktan sonra herhangi bir kesinti olmadıkça çalışmasını sürdürür. Çalışma sırasında yeni bir işe başladığı takdirde, işin sahibi olan kullanıcıya, işe başlandığına ilişkin bir mesaj yollar. Mesaj aşağıdaki formdadır;

MSG FROM BATCHID : JOB 'yourjob' STARTED

Aynı şekilde başlanan işin bitiminde de işin sahibi olan kullanıcıya bittiğine ilişkin bir mesaj yollar. Mesaj aşağıdaki formdadır;

MSG FROM BATHID : JOB 'yourjob' ENDED

Burada kullanılan 'yourjob', kullanıcının /JOB kartında verdiği iş ismidir.

BATCH makinası, okuyucusunda bulunan bir sonraki işe başlamadan önce aşağıdaki işlemleri yerine getirir;

- Tüm havuzlama birimlerini ve sahalarını kapatır,
- CP TAG komutuyla belirtilmiş herhangi bir havuzlama birimini reset eder,
- Bir önceki işde bağlanmış olabilecek tüm diskleri kullanımdan kaldırır,
- Hesaplama birimlerini sisteme yollar,
- CMS'i tekrar yükler.

Yukarıda sözedilen tüm işler, batch makinasının

okuyucusunda bulunan bir sonraki işin çalışması sırasında, bunlardan etkilenmemesi için yapılır. Çalışma sırasında, herhangi bir hatadan dolayı yollanan iş kesilirse, işin sahibi olan kullanıcıya aşağıdaki mesaj yollanır;

MSG FROM BATCHID : JOB 'youjob' ABEND

Ortaya çıkması olası böyle bir durumdan sonra, batch makinası, o andaki dump'ı sistem yazıcısının okuyucusuna yollar. Geri kalan iş silinir ve okuyucuda bulunan bir sonraki işe geçilir. Okuyucuda çalıştırmak için herhangi bir iş bulunmadığı takdirde, gelmesi olası bir iş için beklenir.

9.1 - BATCH İşlemin Hazırlanması

BATCH makinasında çalıştırılması istenilen iş için gerekli olan tüm hazırlıkları, batch makinası içinde uygulanacak şekilde komutları sıralamak gerekmektedir. Programın çalışması sırasında kullanması muhtememl olan tüm disklere bağlantısını sağlamak için gerekli komutların hepsi olmalıdır. Bu, hazırlanan işin başına konacak olan LINK ve ACCESS komutları kullanılarak yapılır. CMSBATCH makinasının, hiçbirşey tanımlanmadığı takdirde, sadece 195 numaralı diski bulunur. Aşağıda verilmiş olan örnek, kendimize ait olan ve varsayılan numarası 291 olan

diskimizdeki ÇALIŞ adlı modülü çalıştırmak için yazılmıştır;

```
/JOB FISH 012345  
CP LINK MCGUIRE 291 291 RR SECRET  
ACCESS 291 B/A  
CALIS
```

Aynı şekilde hem giriş hemde çıkış kütüğü kullanılıyorsa bu durumda hazırlanması gereken iş örneği aşağıdaki gibidir;

```
/JOB FISH 012345  
CP LINK ARDEN 391 391 RR FOREST  
ACCESS 391 B/A  
FILEDEF INFILE DISK VITAL STAT B  
FILEDEF OUTFILE PUNCH  
CP SPOOL PUNCH TO MCGUIRE  
CALIS
```

Yukarıda verilen örnek, aynı zamanda, yollanan programın bir yazıcı çıktısı yaratması gerekiyorsa kullanılması gereken komutları da göstermektedir. Ancak bu şekilde bir tanımlama sadece havuzlama birimini kullanmak için yapılır. Örnekte meydana gelecek olan çıktı MCGUIRE adlı varsayılan makineye yollanacaktır. BATCH makinasında çalışan tüm işlerin yarattığı yazıcı çıktıları, aynı makina tarafından yazıcıya yollanır. Bu kütükler işi yollayan kullanıcı koduyla beraber kullanıcı kodunun sistem direktörüsindeki dağılım kodu adı altında havuzlanır. CP SPOOL komutu yardımıyla bu karakteristikler

değiştirilebilir. Eğer çalışma sonunda meydana gelecek olan çıktının başka bir kullanıcı kodunda havuzlanması istenirse, bu taktirde şu komutu kullanmak gerekir;

CP SPOOL E FOR JONSON

9.2 - BATCH İşlemden CP ve CMS Komutlarındaki Kısıtlamalar

BATCH makinasının çalışması sırasında, CP ve CMS komutlarının bir kısmı desteklenmektedir. Aşağıda verilmiş olan CP komutları, batch sisteminde kullanılabilecek olanlardır;

CHANGE	MSG
CLOSE	QUERY
DETACH	REWIND
DUMP	SMSG
DISPLAY	SPOOL
LINK	STORE
LOADVFCB	TAG

Yukarıda belirtilen komutlarda kullanım açısından bazı kısıtlamalar vardır. Bunlar aşağıda belirtilmiştir;

- CHANGE, SPOOL ve CLOSE komutlarını varsayılan okuyucuda kullanmak mümkün değildir.
- DETACH komutu, herhangi bir havuzlama birimi yada sistem IPL diskini ayırmada kullanılamaz.
- LINK komutu, bir satırda ve aşağıdaki formatta girilmelidir;

CP LINK userid vaddr vaddr mode password
(Link komutunda normal olarak kullanılabilecek olan AS, PASS ve TO kelimeleri batch sisteminde

kullanılamaz. DISK için eğer herhangi bir şifre verilmemişse, ALL olarak şifre seçeneğinde girilmelidir)

- DIAGNOSE 8 kullanıldığı taktirde CHANGE ve SPOOL komutları varsayılan okuyucu üzerinde etki yaparlar.

BATCH işindeki tüm CP komutlarının başında 'CP' karakterlerinin bulunması gerekir. Dikkat edilmesi gereken ikinci bir konuda, eğer CMS EXEC2 veya EXEC, batch makinasında çalışıyorsa, meydana gelebilecek olan hatalarda geri dönüş kodu normal zamanda çalışırken üretilen koddan farklı olabilir. BATCH makinası bazı komutların geçerliliğini incelediğinden, gelen kod, CP nin ürettiği bir kod olmayıp batch makinasına bağlı olan kod olabilir. BATCH makinası okuyucudan işlem gördüğü için aşağıdaki komutlar bu sistemde kullanılamaz;

```
ASSGN SYSXXX READER  
DISK LOAD  
FILEDEF READER  
READCARD
```

Yukarıda verilmiş olan CP komutları gibi bazı CMS komutlarında da aynı türden kısıtlamalar vardır. /SET komutunun aşağıda verilmiş olan operandları geçersiz olanlardır;

ELIP
IMPCP
INPUT
OUTPUT

PROTECT
REDTYPE
RELPAGE

Yukarıda verilmiş olan /SET komutlarının dışında kalan tüm diğer seçenekler kullanılabilir durumdadır. Ancak CP SET komutunun hiçbir seçeneği kullanılamaz.

9.3 - BATCH Özelliğinin Çıktıları

BATCH makinasının çalışması sırasında, çalışan programın oluşturacağı tüm çıktılar işi yollayan kullanıcının koduyla, sistemin yazıcı kuyruğuna havuzlanır. Bu durum, ancak işi yollayan kişinin aksi bir komutu girmediği sürece oluşabilir. Havuzlanmış olan bu kütükler, batch makinasının denetiminden çıkmış olurlar ve burada CP'nin denetimine girerler. Dolayısıyla CLOSE, CHANGE, PURGE, komutları bu kütükler üzerinde çalışır duruma gelmiştir. Okuyucuda bulunan işlerin hangilerininin BATCH makinasına ait olduğunun tesbiti için, batch makinasının ismi, çıktıdaki CP kütük isminde görülür. Kütük tipi ise JOB kartında belirtilendir. Burada birşey belirtilmemişse sadece JOB adını alır. Aynı durum hem yazıcı hemde punch kütükleri için de geçerlidir. Söz

edilen bu kütüklerden ayrı olarak, batch makinasının çalışmasını, meydana gelmiş olan hataların ve daha birçok ayrıntı bilgisinin tutulduğu BATCH CONSOLE adlı bir kütükde oluşturulur. Bu kütük operatöre, meydana gelebilecek olan herhangi bir hata durumunda, hatanın nereden kaynaklandığının tesbiti için kolaylık sağlar.

10 - GERÇEKLEŞTİRİLEN DİZGENİN TANITIMI

10.1 - SİSTEMİN GENEL TANITIMI

Tasarımı ve gerçekleştirimi, sadece local kullanıcılar için tamamlanan dizgemizin iki temel amacı bulunmaktadır. Bunlardan ilki sistemimizde yoğun bir şekilde çalışmakta olan batch işlerin bir sıraya konularak sistemden gereksiz yere vakit çalınmasının önlenmesi, dolayısıyla sisteme gereksiz yere yüklenilmesini ortadan kaldırmak ikincisi ise, sistemde herhangi bir şekilde BATCH özelliğini kullanmak isteyen her kişinin hazırlamak zorunda olduğu iş kartlarının karmaşıklığını ortadan kaldırmayı sağlamaktır.

Tasarımlanan dizge, kullanıcı için söz edilen iş kartlarının hazırlanması işini, kullanıcıya çeşitli sorular sormak ve cevaplara göre çeşitli deyimleri alt alta getirmek şeklinde yerine getirir. Meydana getirilen bu iş kartları yığını, sistemde bulunan BATCH makinasına yollanır, ortaya çıkacak olan herhangi bir sonuç varsa, bunlar direk olarak işi yollayan kullanıcıya geri yollanır.

Dizgemizin gerçekleştiriminde, çeşitli aritmetiksel ve karakter fonksiyonlar üzerinde etkin kullanılabilirliğinden dolayı, programlama dili olarak PL/1 seçilmiştir.

Ayrıca kullanıcı ile olan etkileşimde ise DMS (Display Management System) kullanılmıştır. Bu sayede kullanıcı ile etkileşim, çok kolay bir şekilde sağlanabilmiş ve veriler üzerinde çok rahat işlem yapabilme olanağı sağlanmıştır.

Kullanıcı herhangi bir şekilde batch sistemini kullanmaya çalışırsa, öncelikle, sistem sorumlusundan alınabilecek olan, şifre kelime sorulur. Bu sorguya doğru yanıt verebilme hakkı sadece üç defa olarak kabul edilmiştir. Üç haktan üçünde de yanlış şifre girildiği takdirde, sistem kullanıcıya bir mesajla bu olayı bildirir ve sistem sorumlusu ile görüşmesi ikaz edilir. Şifre doğru girildiği takdirde dizgenin ana menüsü ekrana yansır. Bu menüde kullanıcının ne tür bir iş yapmak istediği sorgulanır. Tasarımlanan dizgede, kullanıcının sadece DERLEME ve PROGRAM ÇALIŞTIRMA gibi iki seçeneği bulunmaktadır. Tasarım amacına bakılacak olursa, bu iki seçenek yeterli görülmektedir. Ayrıca kullanıcı için yardım seçeneği bulunmaktadır. Bu seçenek seçildiği takdirde, ilgili menüde ne yapılması gerektiği açık bir şekilde anlatılmıştır. Yine de anlaşılınmayan bir durum olduğu takdirde sistem sorumlusu ile görüşmek mümkündür.

10.2 - SİSTEMİN TANITIMI

Planlanan ve gerçekleştirilen sistemin temel amacı kullanıcıyı CNSBATCH makinasını kullanabilmek için

hazırlanması gereken JCL kartlarından kurtarmaktır. Karşılıklı iletişimli olarak sorulan sorulara, yapılması istenen işleme göre verilecek yanıtlar doğrultusunda JCL kartları hazırlanır. Kullanıcıya sistemde **YARDIM** olanağında sunulmuştur. Bu sayede kullanıcı herhangi bir anda ne yapması gerektiğini genel olarak bu yardım olanağından kolayca okuyup anlayabilir. Kullanıcının sistemden yaralanabilmesi için öncelikle kullanıcı kodunu ve CMSEATCH makinasının şifresini doğru yazması gerekir. Bunun sebebi, batch makinasında çalışacak olan tüm işler sistemden bir zaman çalacaklardır, bu zamanın kullanıcılara para olarak yansıtılması istendiği durumda CMSEATCH makinasının kimlere hizmet verdiği hatasız bulunabilmesidir. Şifrenin yanlış yazılması durumunda sistem kullanıcıya üç hak tanımaktadır. Üçüncü haktan sonra önce sistem sorumlusuna olmak üzere sistem operatörünede ilgili mesajı yollar. Sistem ilk tasarlandığı haliyle sadece program derleme ve çalıştırma ile ilgili olan JCL kartlarının hazırlanmasında yardımcı olmaktadır. Uzak birimlere iş yollamak ve sonucunu almak ile ilgili olarak hazırlanması gereken JCL kartlarının ve sistemlerde olması gereken VM'lerin neler olduğu (Bölüm 10.2.4) 'de açıklanmıştır. Bu aşamada, program derleme ve

çalıştırma seçeneklerinden yararlanırken sistemin çalışması, bu çalışma sırasında kullandığı dizinler, bunların yapıları, menüler ve bunların açıklamaları üzerinde durulmuştur.

10.2.1 - SİSTEMİN ÇALIŞTIRILMASI

Sistemin tüm kullanıcıların ortak olarak kullandıkları ve sistem yazılımlarının bulunduğu diske yerleştirilmesi planlanan CM2 EXEC programı ile çalıştırılması olasıdır. Bu programın ayrıntılı bir akış şeması (Ek A1 ve A2) 'de verilmiştir. Sistemin başlangıç modülü olduğundan, sistemin çalışması sırasında kullanıcının isteyebileceği yardım olanağının kütük tanımlaması, ilk olarak yapılır. Aynı modülde sistemin çalışması sırasında oluşturması istenen JCL kartlarının kayıt edileceği kütük tanımlamasıda yapılır. Her iki kütüğün ayrıntılı açıklamaları (Bölüm 10.2.3)'de verilmiştir. Bu aşamada sonra sistemi kullanmak isteyen kullanıcının kodunun bulunması gelir. Bu işlemi ise FILE USER isimli kütüğün olup olmadığını kontrol ederek yapar. Bu isimde bir kütük yoksa aynı isimde bir kütük yaratıp kullanıcının kodunu, sistemin daha ileriki aşamalarda kullanabilmesi için kayıt eder. Sistemden bulunabilen bu kodun kullanıcı tarafından tekrar girilmeye zorlanmasının

sebebi, hem kullanıcıya bir güven hissi vermek hemde bilerek yanlış girme olaylarını ortadan kaldırmaktır. Ayrıca çifte güvenlik sağlamaktadır. **FILE USER** kütüğünün oluşturulmasından sonra sistemin sonraki aşamadaki modülü çalıştırılır. Bu aşamadaki işlemler ayrıntılı olarak (Bölüm 10.2.2)'de anlatılmıştır. Bu modülün çalışması bittikten ve kullanıcı JCL kart hazırlamasının bittiğini belirtmesinden sonra aynı program bazı kontrolleride gerçekleştirir. Bunlar CMSBATCH sisteminin daha sağlıklı çalışmasını sağlamak için derleme aşamasında, derlenmesi istenen programın olup olmadığı, çalıştırma aşamasında ise hem çalıştırılacak programın hemde bunun kullanacağı kütüklerin olup olmadığının kontrolleridir. Bu kontrollerden sağlıklı geçilirse, kullanıcının hazırlamış olduğu JCL kartları sisteme yollanır. Bu anda kullanıcının sistem ile olan ilişkisi bitmiş olmaktadır, bu yüzden çalışma kütüğü olarak kullanılan sahalara bir dahaki kullanımlar için silinir.

10.2.2 - MENÜLER ve AÇIKLAMALARI

Sistemin çalışması sırasında kullanıcı ile olması gereken karşılıklı iletişimli ortam, DMS (Display Management System) hazır yordamı kullanılarak sağlanmaktadır.

10.2.2.1 - BATCH0 MENÜSÜ

Sistemin başlangıç menüsüdür. Sistemin, çalışmaya başladığı zaman karşımıza gelen bu menüde, kullanıcıya iki seçenek sunulmuştur (Örnek menü (Ek B)'de verilmiştir). Bunlar ,program derleme ve program çalıştırma seçenekleridir. Program derleme seçildiği zaman kullanıcının yaklaşık on ayrı programlama dili yada istatistik paketini kullanmak istediği anlaşılır. Daha sonraki menülerde ise kullanıcıya sorulacak olan sorulara vereceği yanıtlara göre JCL hazırlama işlemi devam eder. Bu menüde program çalıştırma seçilirse yine on ayrı programlama dili yada istatistik paketi komutları ile yazılmış olan programların çalıştırma isteği sisteme iletilmiş olur. Daha sonraki menülerde kullanıcının bu istemini yerine getirebilmek için çeşitli sorular sorulur, bunlara alınacak olan yanıtlara göre JCL hazırlanıp batch makinasına yollanır duruma getirilir. Bu menüde PF1 tuşuna basıldığında kullanıcıya BATCH0 menüsü ile ilgili olarak bir yardım ekranı gelir. PF3 tuşuna basıldığında sistemin çalışması tamamlanır ve çalışma sırasında oluşturulmuş olan JCL'ler batch makinasına yollanır.

10.2.2.2 - BATCH1 MENÜSÜ

Çalışma sırasında kullanılacak olan veri

kütüklerinin, çalışma sahalarının görüntü adreslerini ve hangi programlama dilinin kullanıldığının belirtileceği menüdür (örnek bir menü (Ek B)'de verilmiştir). Burada kullanıcı kendi kullanıcı kodunu doğru girmek durumundadır. Sistemin çalışması bu kodun doğru girilmesine bağlıdır. Ayrıca sistem zamanının kimin tarafından kullanıldığının bulunması açısından da önemlidir. Kullanıcı kodunun girilmesinden sonra CMSBATCH anahtar sözcüğünün de doğru girilmesi gerekmektedir. Bu şifre sistem programcılarında öğrenilebilir. Şifrenin üç kez yanlış girilmesi durumunda sistemin çalışması kesilir ve durum sistem programcı ve operatörüne bildirilir. Şifrenin doğru girilmesi durumunda BATCH0 menüsündeki seçime bağlı olarak derlenecek yada çalıştırılacak program için gerekli olan disklerin adresleri ve bağlantı şifre kelimeleri girilmelidir. BATCH0 menüsünde derleme seçilmişse, derlenmesi gereken programın hangi görüntü diskte bulunduğu ve ilgili diskin bağlantı şifresinin girilmesi gerekmektedir. BATCH0 menüsünde çalıştırma seçeneği seçilmişse, hem programın bulunduğu disk adresini ve şifresini hemde çalışması sırasında kullanması gereken disklerin adres ve şifrelerinin girilmesi zorunludur. Son

olarak yine BATCH0 menüsünde seçilmiş olan seçeneğe bağlı olarak hangi programlama dilinde işlem yapıldığının bildirilmesi gerekir. Bu menüde PF1 tuşu yardım fonksiyonunu çağırır ve ilgili menü ile yardım ekranı gelir. PF3 tuşu BATCH0 menüsüne dönüş içindir. Bu menüde ilgili tüm alanların doldurulmasından ve ENTER tuşuna basılmasından sonra bir sonraki menüye geçiş sağlanır. Bu geçiş sırasında kullanıcının şimdiye kadar girmiş olduğu bilgilere dayanarak şu JCL'ler hazırlanmış olur (Kullanıcının girmiş olduğunu varsaydığımız kelimeler daha koyu yazılmıştır);

```
CP SPOOL PUNCH TO CMSBATCH CONT
&PUNCH /JOB CMSBATCH CENGİZ LEVENT
&PUNCH ACCESS 195 A
&PUNCH CP LINK SYSADMX 191 500 RR CENGİZ
&PUNCH ACCESS 500 B
&PUNCH ACCESS 319 P
&PUNCH ACCESS 19E Y
&PUNCH ACCESS 326 Q
&PUNCH GLOBAL MACLIB CMSLIB
&PUNCH GLOBAL TXTLIB CMSLIB PLILIB
```

10.2.2.3 - BATCH2 MENÜSÜ

Kullanıcı bu menüde, derlemek yada çalıştırmak istediği program ismini girmek zorundadır (Örnek bir menü (Ek B)'de verilmiştir). Herhangibir ihtiyaç karşısında YARDIM olanağı PF1 tuşu ile sağlanmıştır. PF3 tuşu bir önceki menüye dönmek için kullanılır. Kullanıcının seçeneklerine bağlı olarak bu bölümde hazırlanacak olan

JCL kartlarına örnek aşağıda verilmiştir (Kullanıcının sistemi kullanması sırasında girdiği değerler daha koyu belirtilmiştir);

- 1- &PUNCH PLIOPT CENGİZ
- 2- &PUNCH CP SPOOL PUNCH TO SYSADMX
- 3- &PUNCH DISK DUMP CENGİZ LISTING
- 4- &PUNCH /*
- 5- CP SPOOL PUNCH NOCONT CLOSE
- 6- &EXIT

Program derleme menüsünün seçilmiş olduğunu kabul ettiğimizden, kullanıcının bu aşamadan sonra herhangi bir işlem yapmasına gerek kalmamaktadır. Bu yüzden işin bittiği ve başka bir işe başlaması gerektiğinin batch makinasına bildirilmesi lazımdır. Bu işlem dördüncü satırdaki işlem ile olmaktadır. İkinci ve altıncı komutlar, ilgili işin bitmesi ve sonuçlarının işi yollayan makinalara yollanmasını sağlamak içindir. Birinci komut ise programın derlenmesi için gerekli olan komuttur. Yukarıdaki örnek program derleme ile ilgilidir. Çalıştırma ile ilgili olarak (Bölüm 10.2.2.4)'de bir örnek verilmiştir.

10.2.2.4 - BATCH3 MENÜSÜ

Sistem bu menüye BATCH0 menüsünde program çalıştırma seçilmişse geçer (Örnek menü (EK B)'de verilmiştir). Kullanıcı burada programın çalışması sırasında kullanacağı

mantıksal ve fiziksel kütük isimlerini girmek zorundadır. Programın kullandığı birden çok kütük tanımı için kullanıcı PF12 tuşunu kullanabilir. PF12 tuşuna basıldığında daha önce sahalarda bulunan bilgiler saklanıp silinecek ve ekranda sahalar temizlenecektir. Programın oluşturması muhtemel bir çıktı kütüğünü, OUTPUT seçeneğini seçerek yapabilir. Bu şekilde, oluşacak olan çıktı kütükleri sistemin çalışmasından sonra kullanıcıya geri yollanmasını sağlar. Kullanıcının OUTPUT bilgisine herhangi bir şey girmediği tüm kütükler INPUT kütüğü olarak kabul edilir. Bu menüde kullanıcının girmiş olduğu bilgiler ile ilgili örnek aşağıda verilmiştir;

Kullanıcının giriş kütüğü olarak fiziksel ismi YARDIM DATA, mantıksal ismi CENGİZ ve boyu 70 olarak belirttiğini düşünelim, ilgili tanımlama şu şekilde olacaktır;

```
&PUNCH FILEDEF CENGİZ DISK YARDIM DATA A
      (RECFM F LRECL 70
```

Kullanıcı programda kullandığı çıktı kütüğünün tanımını da şu şekilde yapmış olduğunu düşünelim;

```
&PUNCH FILEDEF CIKTI DISK CIKTI DATA A
      (RECFM F LRECL 100
```

Bu iki tanımlamadan sonra sistem diğer kartları otomatik olarak oluşturmaktadır;

```
&PUNCH LOAD CENGİZ
&PUNCH START
&PUNCH CP SPOOL PUNCH TO SYSADMX
&PUNCH DISK DUMP CIKTI DATA
&PUNCH /*
CP SPOOL PUNCH NOCONT CLOSE
&EXIT
```

Derleme ve çalıştırma seçeneklerinde hazırlanan JCL kartlarına birer örnek (EK C)'de verilmiştir.

10.2.3 - KULLANILAN KÜTÜKLER, AÇIKLAMALARI VE AMAÇLARI

10.2.3.1 - FILE USER A

Sistem çalışmaya başladığı anda oluşturulur. Burada kullanıcının kullanım kodu sekiz karakter olarak tutulur. Sistemin çalışması sırasında çeşitli menülerde istenen kullanıcı kodunun burada yazılmış olanla aynı olması gerekmektedir. Bu kontrolün program içinden yapılmasını sağlamak için kullanılır. Kütük sadece bir kayıttan oluşmaktadır. Kütük yapısı (Şekil 10.1)'de verilmiştir.

```
+-----+      Kayıt uzunluğu...: 8  
IXXXXXXXXXI    Bulunduğu disk...: 191 A  
+-----+      Tipi.....: sıradan
```

FILE USER KÜTÜK FORMATI ŞEKİL 10.1

Sistemin çalışması bittikten sonra bu kütük silinir. Böylece kullanıcının bu kütüğü editleyerek değiştirme fırsatı verilmemiş olur.

10.2.3.2 - CM EXEC

Sistemin çalışması sırasında hazırlanan JCL deyimlerinin yazıldığı kütüktür. Sistem

yorumlayıcılarından birisi ile çalıştırılması gerektiğinden tipi EXEC olarak seçilmiştir. Sistemin ana modülü olan CM2 EXEC (Bölüm 10.2.4)'de açıklanmıştır), kullanıcının JCL yaratmak ile ilgili olan işlemi bittiği zaman bu kütüğü çalıştırır. Bu kütükte bulunan JCL'lerde CMSBATCH sistemine iş yollamak deyimlerini içerdiğinden, kullanıcının yapmak istediği işlem başlatılmış olur. Tanımlanan diskler ve yapılması istenen işlemler CMSBATCH makinasında çalışmaya başlar. Sonuçları ise, işlemin bitiminden sonra işi yollayan kullanıcıya geri yollanır. Kütük yapısı ise (Şekil 10.2)'de verilmiştir.

```
+-----+
I  jcl deyimleri                               I
+-----+
```

Kayıt uzunluğu.....: 80
Bulunduğu disk.....: 191 A
Tipi.....: sıradan

CM EXEC KÜTÜK FORMATI
ŞEKİL 10.2

10.2.3.3 - FILE HELP

Kullanıcının sistemi kullanması sırasında karşılaşılabileceği herhangi bir sorunu ortadan kaldırmak için kullanıcıya online yardım sağlamada kullanılır. Bu gibi bir durumda kullanıcı hangi ekranda olursa olsun PF1 tuşuna basması yeterlidir. Kullanıcıya o anda bulunduğu

ekranla ilgili olarak çeşitli bilgiler gelecektir. Bu kütükte iki ayrı yardım açıklaması arasını '*' karakterleri ayırmaktadır. Yardım açıklaması başlangıcı ise önceden belirlenmiştir. Herhangibir istek karşısında ilgili değişkenler ayarlanıp ilgili yordama sapılır. Bu akışla ilgili olarak ayrıntılı bir şema (EK A)'da verilmiştir. Yardım kütüğünün yapısı (Şekil 10.3)'de gösterildiği gibidir.

```
+-----+-----+-----+
I key   I açıklama                               I
+-----+-----+-----+
```

```
kayıt uzunluğu...: 75
bulunduğu disk...: 326 Q
tipi.....: doğrudan
```

FILE HELP KİTİK FORMATI
ŞEKİL 10.3

10.2.4 - KULLANILAN EXEC KÜTÜKLERİNİN AÇIKLANMASI

10.2.4.1 - CM2 EXEC

Sistemin ana modülünü oluşturmaktadır. Yardım kütük tanımlamaları, sistemin çalışmasında oluşacak olan JCL kütük tanımlaması ve kullanıcı kodunun tutulduğu kütük tanımlaması burada yapılır. Daha sonra sistemin diğer modüllerinin çalıştırılmasına geçilir. CM2 EXEC (EK F)'de verilmiştir.

10.2.4.2 - BATYIL EXEC

Bu modül sistemin çalıştırılması sırasında oluşmaz. Bu çalışma içerisinde, böyle bir imkanında olduğunu göstermek için eklenmiştir. Daha ileriki aşamalarda bu sistemde uzak sistemler için JCL hazırlayan bölümleri hazırlanabilir. Bu modül uzak sistemlere herhangi bir iş yollamak istendiği zaman kullanılabilir. Bu modül, yollanmak istenen işin hangi uzak sisteme yollanacağı ile ilgili tanımlamayı yapar. Bu özelliğin kullanılabilmesi için sistemde mutlaka RSCS (Remote Spooling Communication Subsystem) makinasının çalışır olması gerekir. Yapılan tanımlamada RSCS makinası ile ilgilidir. BATYIL EXEC (Ek D)'de verilmiştir.

10.2.4.3 - YILDIZ EXEC

Bu modül de mevcut sistemde çalışır halde değildir, ancak örnek olarak hazırlanmıştır. Bu modülde uzak sisteme yollanacak olan işin hazırlanmış şekli bulunur. (Ek E)'de verilmiş olan örnek hem program derleme hemde çalıştırmayı içerir. (Bölüm 10.2.4.2)'de verilmiş olan BATYIL EXEC ve bu bölümde verilmiş olan YILDIZ EXEC, İstanbul YILDIZ Üniversitesi ile denenmiş ve sonuç başarılı olmuştur. Programın derlenmesi ve daha sonra çalıştığı zaman oluşturduğu küçük başarı ile geri alınmıştır.

11 - SONUÇ

Çalışmamızda, sistemelerde bulunabilecek olan batch özelliğinin local olarak çalışmasında, kullanıcılar ile batch makinası arasında arabirim tasarımı gerçekleştirilmiştir. Sistem başlangıç olarak sadece yerel kullanıcılar arasında kullanılabilir durumdadır. Bundan sonra yapılabilecek olan çalışma, bu sistemin uzak sistemlerde hizmet verebilecek şekilde geliştirilmesi olmalıdır.

Sistem, PL/I programlama dilini, bunun yanında ekran düzenleme kolaylıklarından dolayı, DMS (Display Management System) yazılımını kullanmaktadır. Tasarımda, yanıt süresinin arttırılmasını sağlamak için ASSEMBLER yada sistem yorumlayıcılarından biri olan REXX, kullanılabilir.

Tasarımlanan ve gerçekleştirilen sistemin, uzak sistemler tarafından da kullanılacak şekilde geliştirilmesi en büyük dileğimizdir.

ÖZET

Herhangibir sistemde bulunması muhtemel olan batch makinası ile kullanıcılar arasındaki iletişimin yalınlaştırılmasını sağlamak için bir sistem gerçekleştirmesi üzerinde çalışılmıştır. Programlama dili olarak PL/I, yardımcı ürün olarak da DMS (Display Management System) kullanılmıştır.

Üzerinde çalışılmış olan sistemin yapısı gereği, ilk bölümde, online ve batch sistemler arasındaki temel farklılıklar üzerinde durulmuş ve bazı örnekler verilmiştir. İkinci olarak, tasarımı gerçekleştireceğimiz ortamın tanıtılması üzerinde durulmuş, varsayılan bellek ve bu mantık açıklanmaya çalışılmıştır. Daha sonra ara birimi tasarlayacağımız CMSBATCH sisteminin çalışması ve altında yatan mantığın geniş açıklaması bulunmaktadır. Sistemin çalışması sırasında yapılması gerekenler, sistem programcısının, sistem operatörünün ve kullanıcıların yapmaları gerekenler geniş bir şekilde anlatılmıştır. Son olarak da, tasarımın çalışması sırasında oluşacak çıktı örnekleri ve bu işlerin nasıl hazırlandığı üzerinde durulmuş, ayrıca tasarımılanan dizge hakkında bilgi verilmiştir.

SUMMARY

In this project, we tried to simplify the communication between the user and batch system by designing an interface between them. We used PL/I as a programming language and DMS (Display Management System) for helping us in designing the interactive screens.

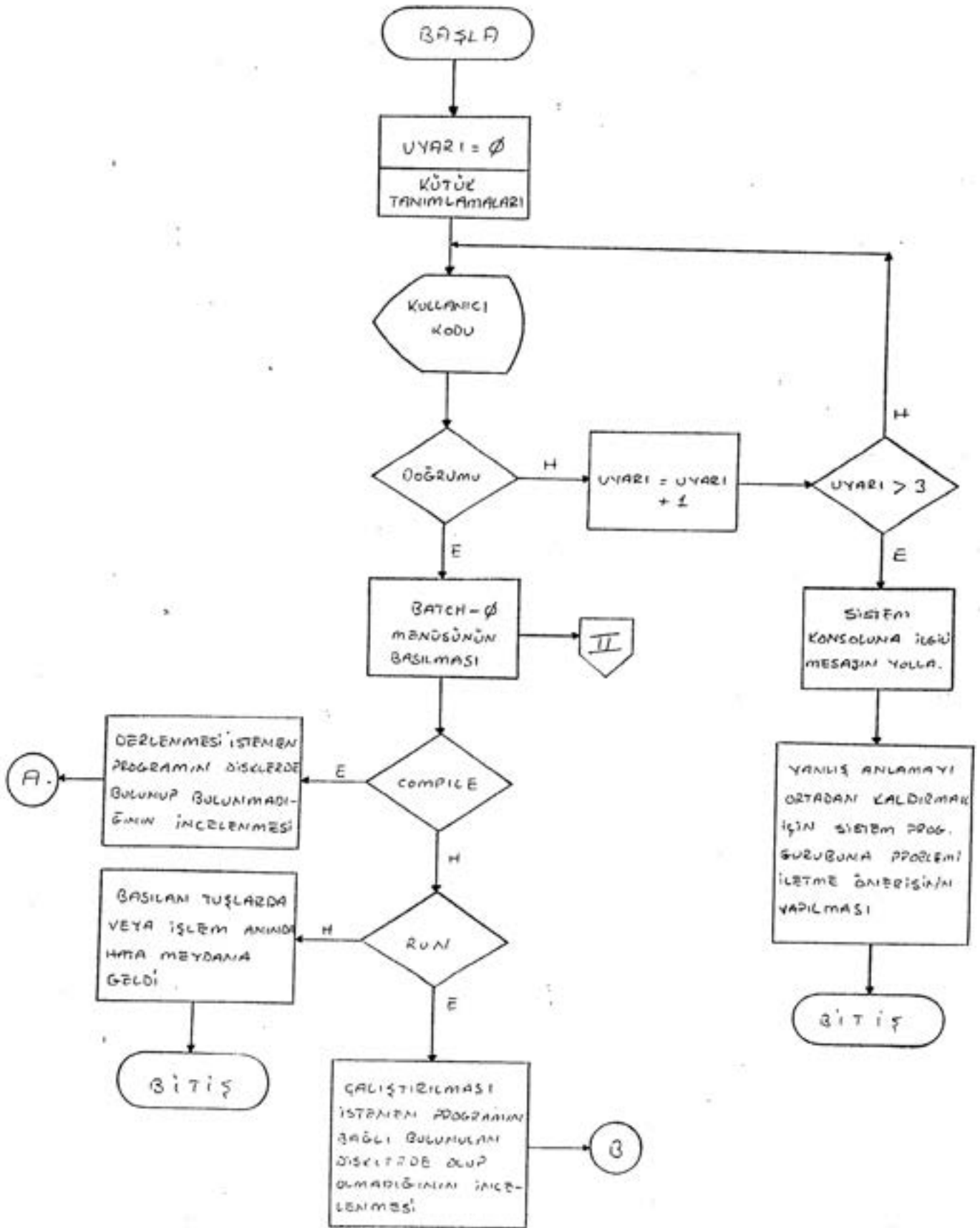
Because of the batch system we worked on, we studied the differences between the online and batch systems and their logic, in the first chapter. In the next chapter, we explained and studied the system that we made our project work on. Virtual memory and its logic explained in detail here.

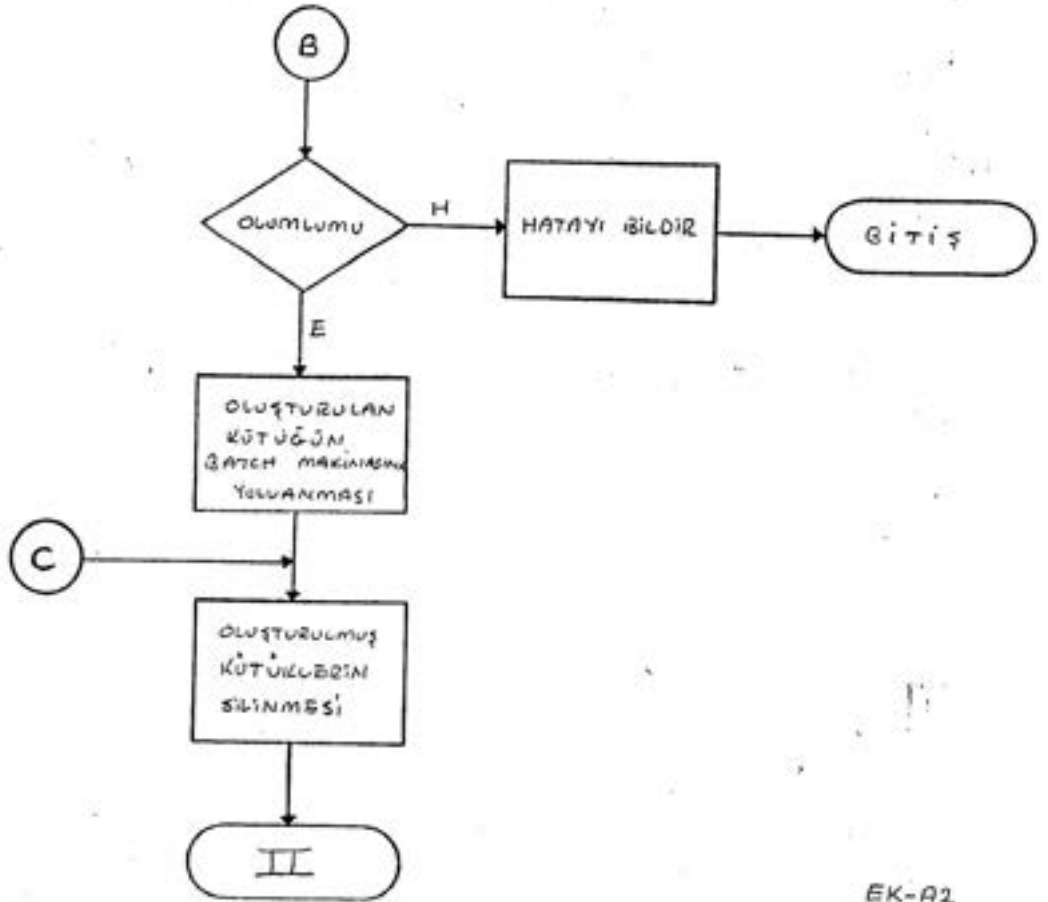
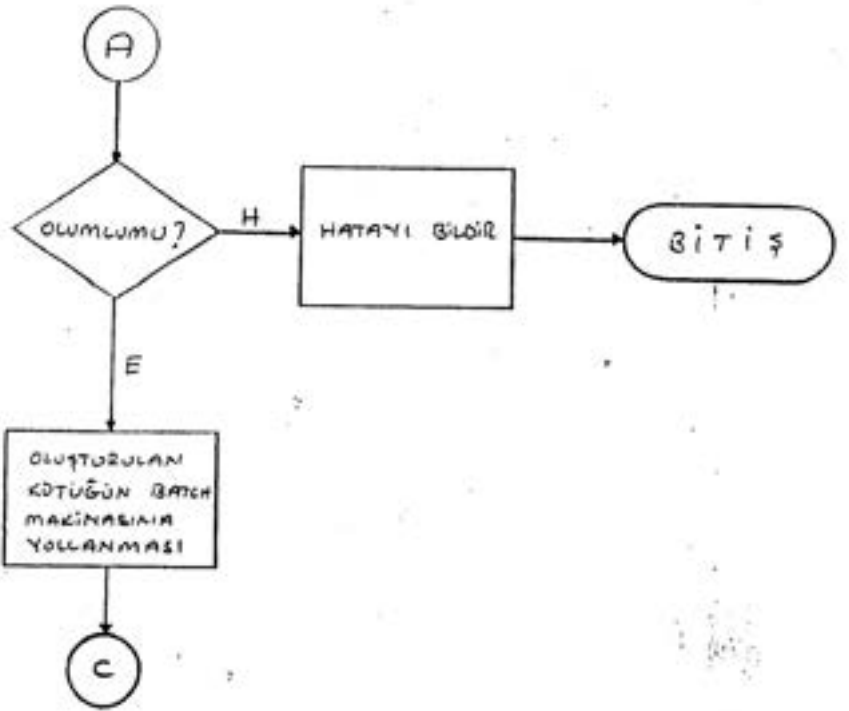
After these explanations we discussed the CMSBATCH system and its working strategies. During the system's work, things that have to be done by the system programmer, system operator and user explained in the next chapter. Finally, the project that we studied on discussed and some output's has been given as an example.

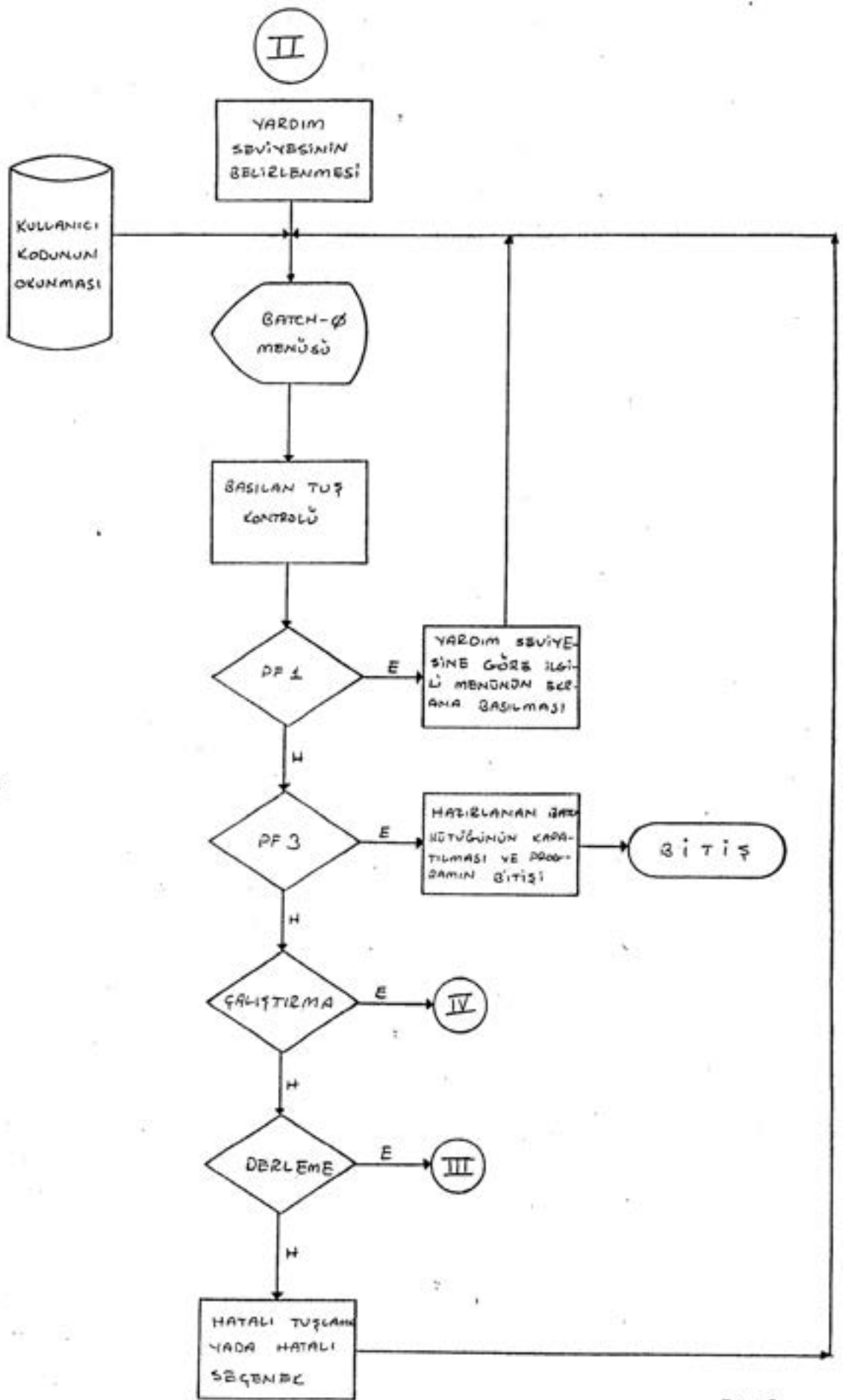
LİTERATÜR LİSTESİ

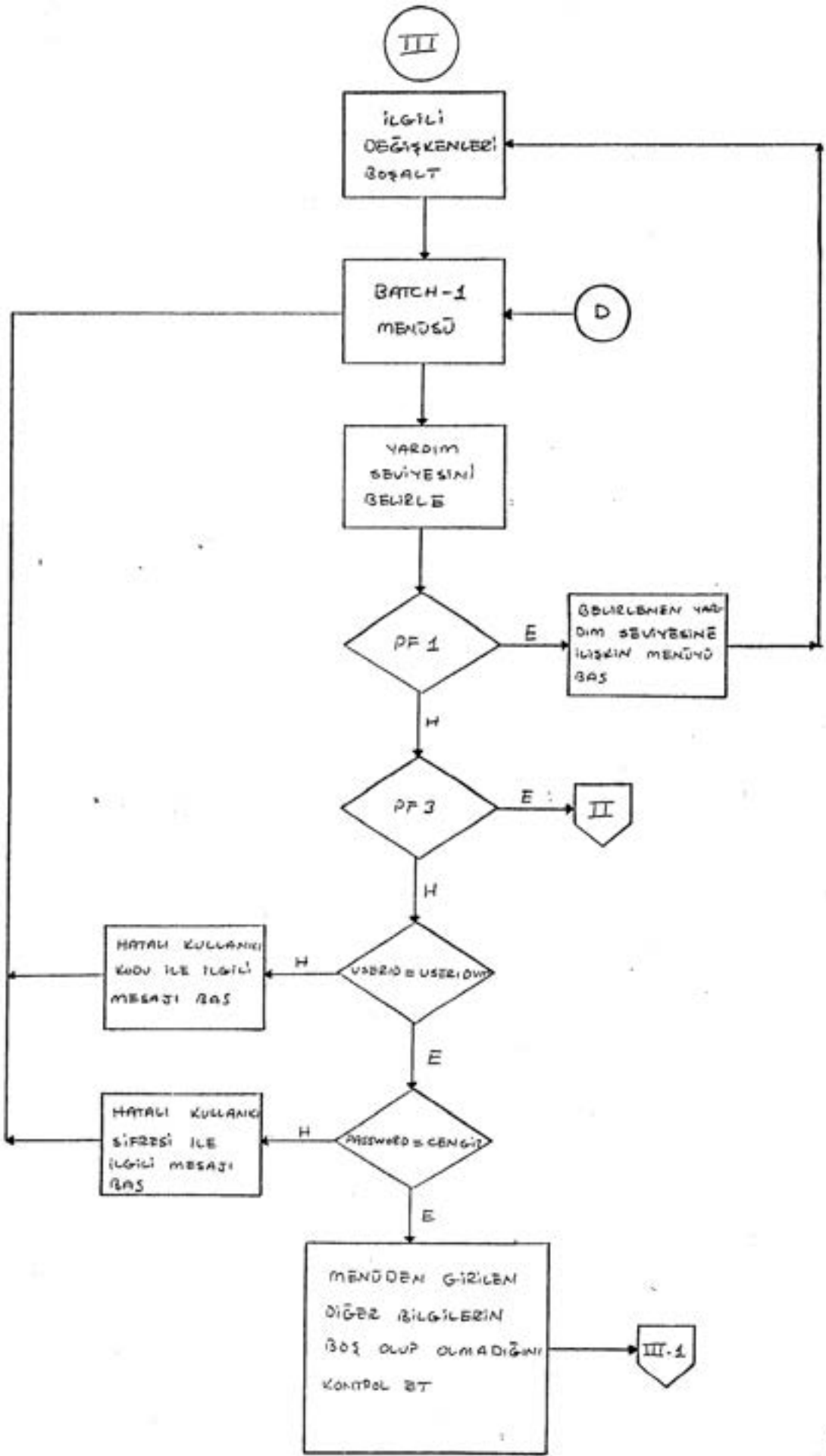
- ALAN, C.SHAW 1974, THE LOGICAL DESIGN OF OPERATING SYSTEMS, PRENTICE-HALL, sf:2
- ÇETİN, REHA 1988, BİLGİSAYAR SİSTEMLERİNDE PERFORMANS İLE DARBOĞAZLAR ARASINDAKİ İLİŞKİLERİN SAPTANMASI ÜZERİNE BİR ARAŞTIRMA, EGE ÜNİVERSİTESİ, İZMİR.
- DONOVAN, JOHN 1972, SYSTEM PROGRAMMING, MCGRAW HILL, sf: 376 , 378
- GORDON, B.DAVIS 1982, INTRODUCTION TO COMPUTERS, MCGRAW HILL, sf:35-38 , 66-68
- HAMACHER V.C. , Z.G.VRANESIC AND S.G.ZAKY, 1978, COMPUTER ORGANIZATION, MCGRAW HILL, sf:311 , 313
- IBM, 1984, CMS USER'S GUIDE, IBM VIRTUAL MACHINE/SYSTEM PRODUCT REL-4, sf:12.1 , 12.5
- IBM, 1983, GENERAL INFORMATION, IBM VIRTUAL MACHINE/SYSTEM PRODUCT REL-3, sf:4 , 12-17
- IBM, 1984, INTRODUCTION, IBM VIRTUAL MACHINE/SYSTEM PRODUCT REL-4, sf:61 , 64
- IBM, 1984, OPERATOR'S GUIDE, IBM VIRTUAL MACHINE/SYSTEM PRODUCT REL-4, sf:5.12 , 5.16
- IBM, 1984, SYSTEM PROGRAMERS GUIDE, IBM VIRTUAL MACHINE/SYSTEM PRODUCT REL-4, sf:519 , 522

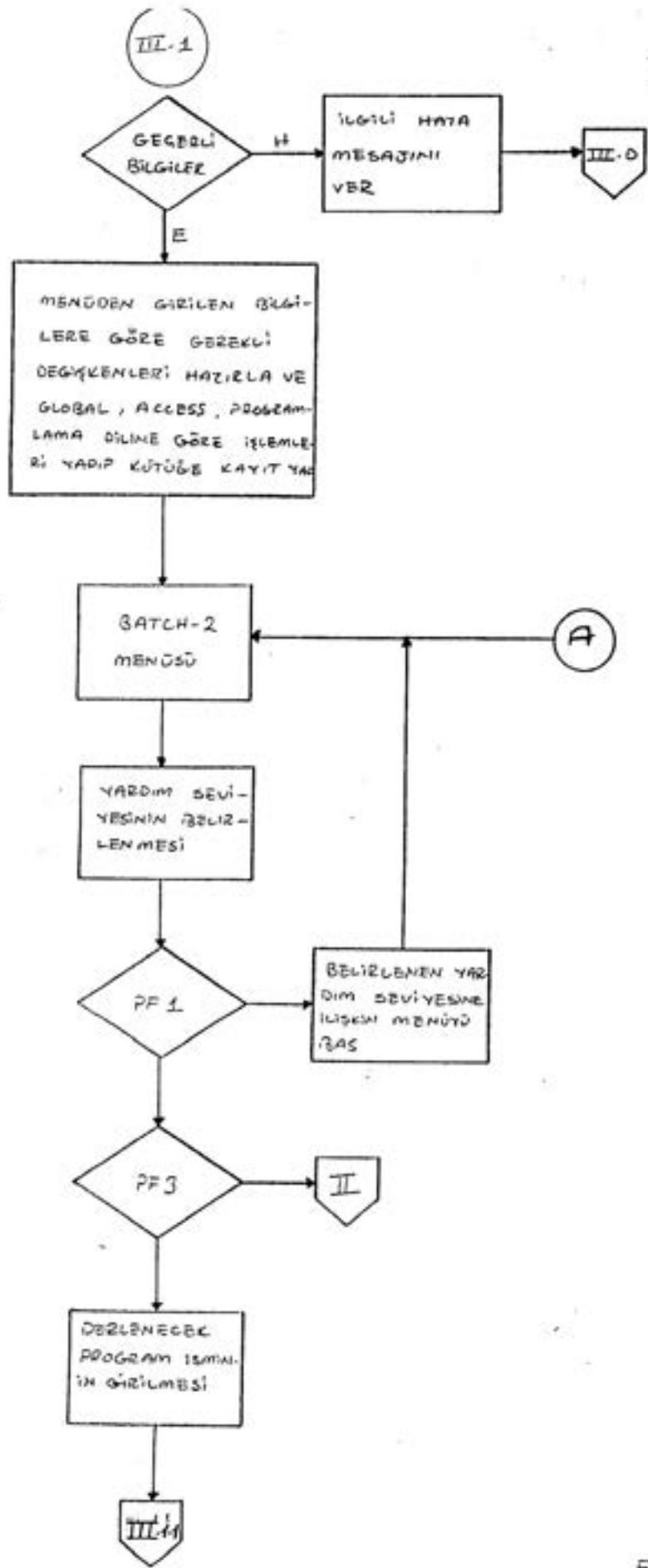
EKLER



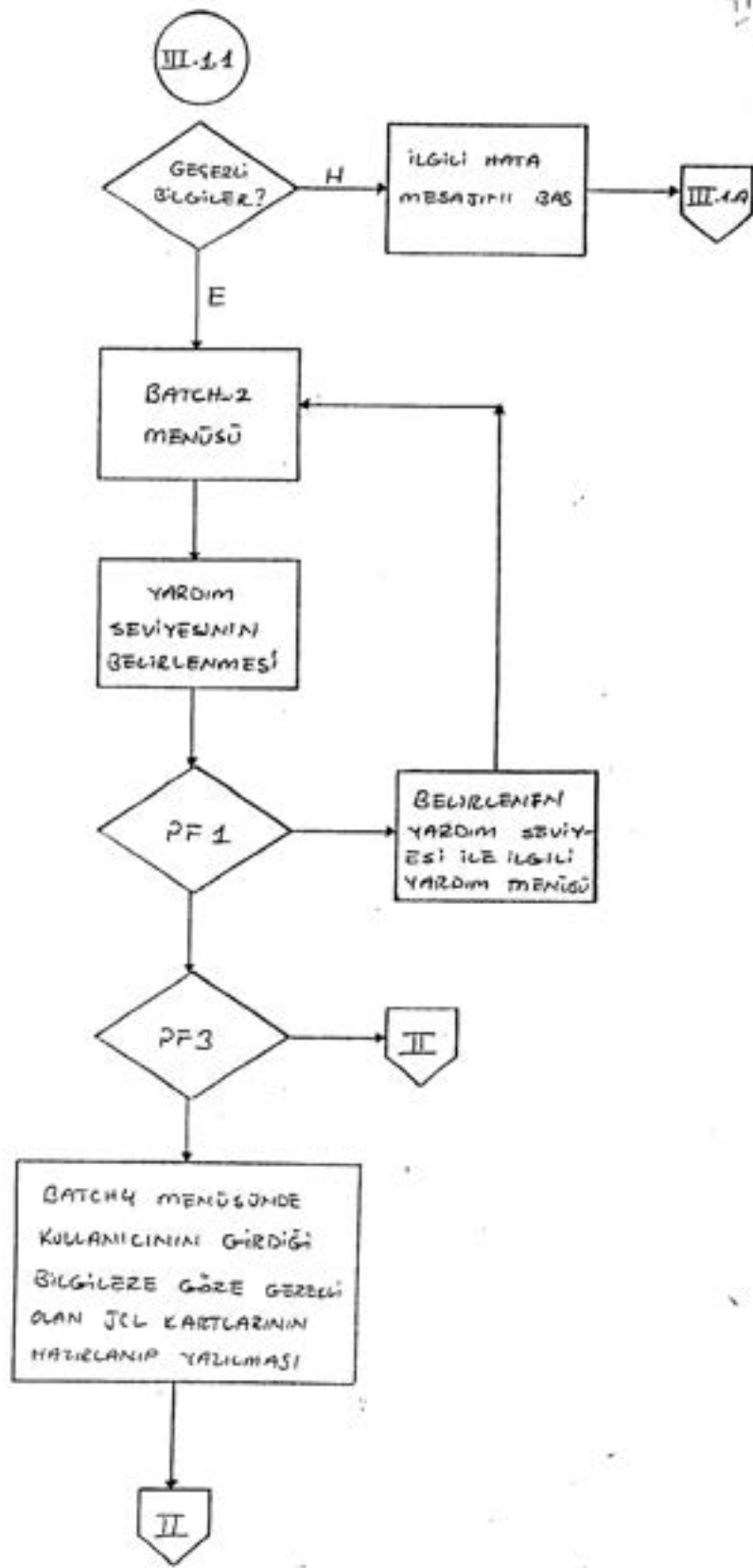




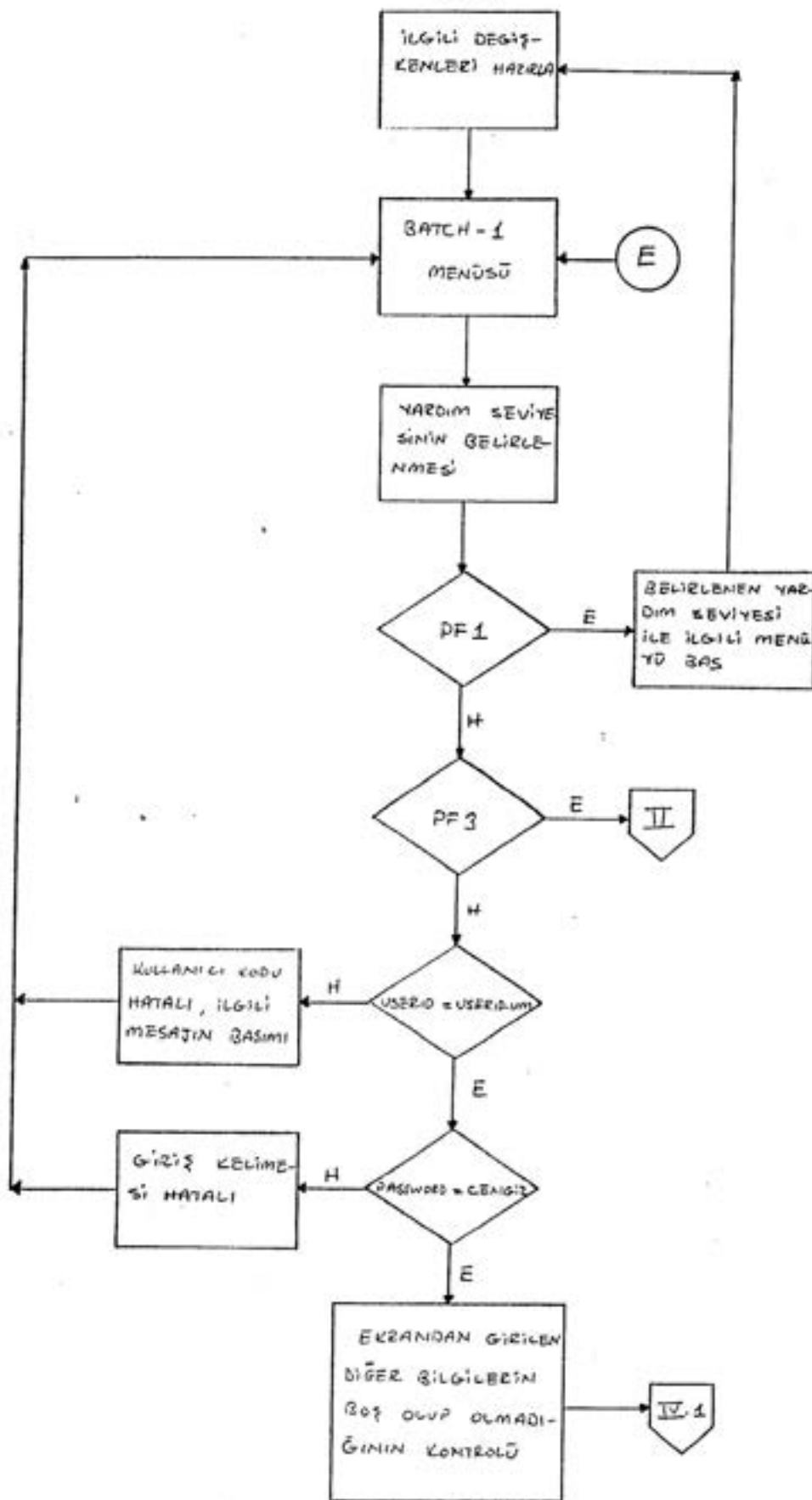




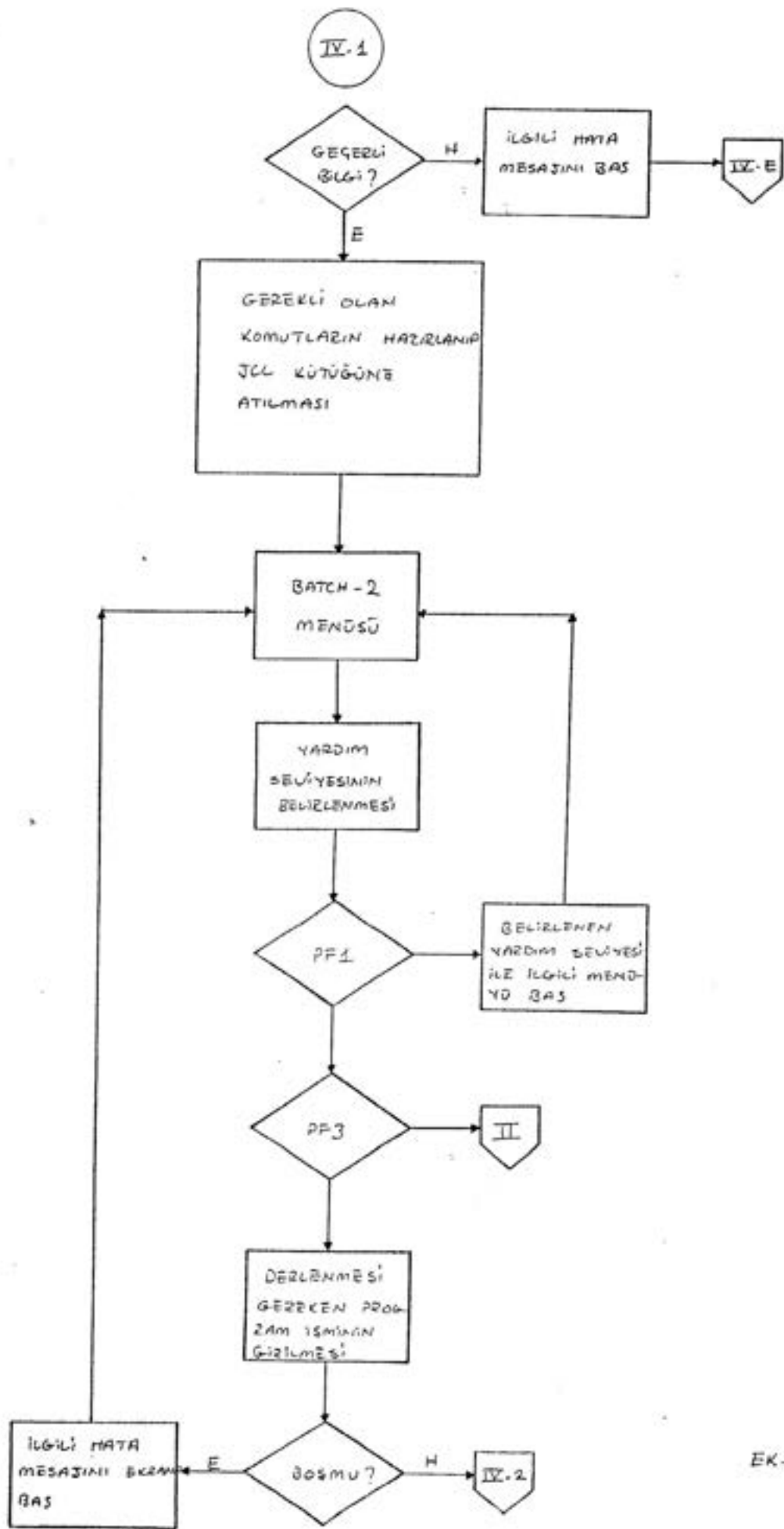
III-1



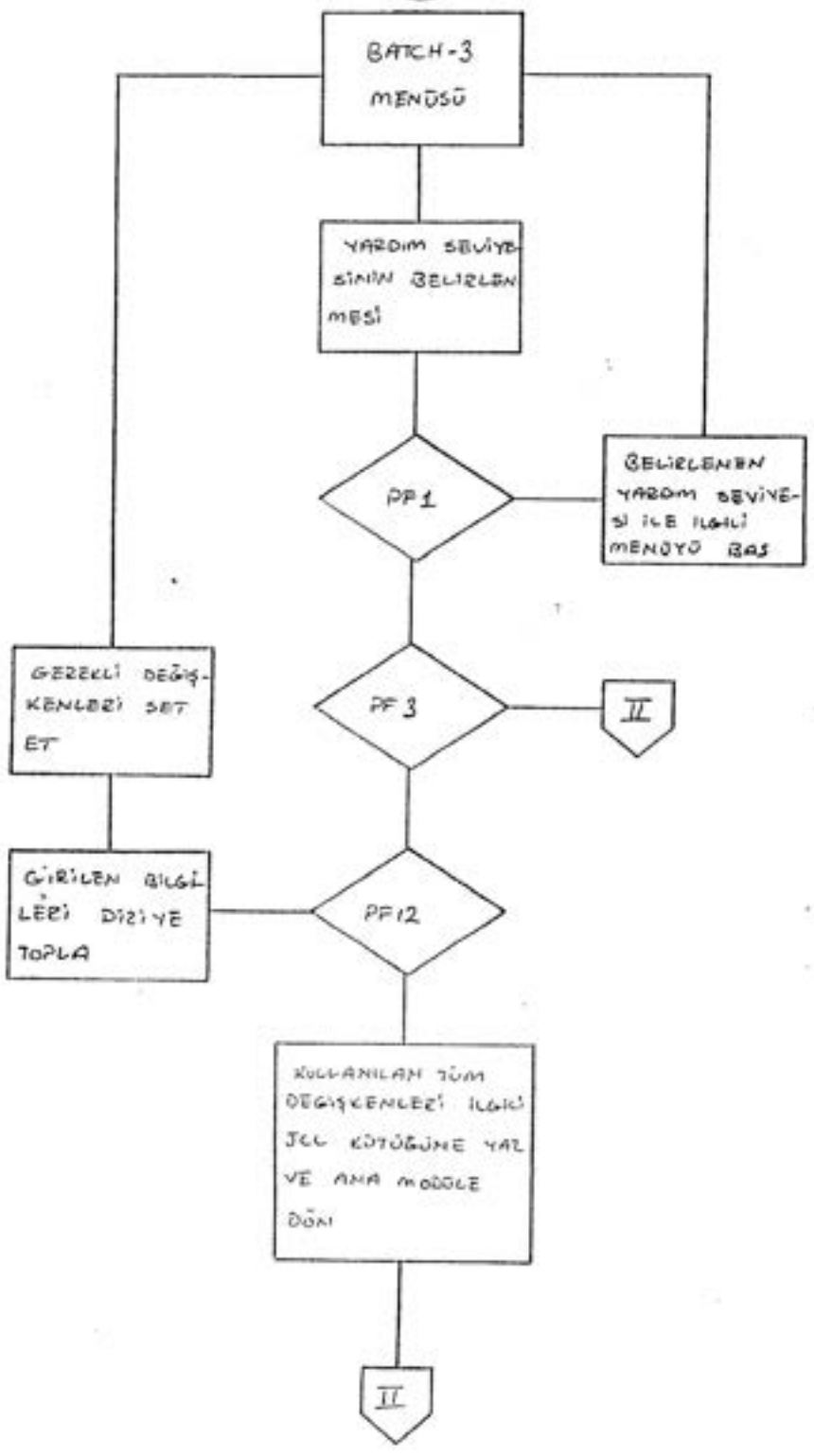
IV



IV.1



IV.2



ANA MENÜ

```
=====
I PROGRAM DERLEME.....: [ ] I
I
I PROGRAM ÇALIŞTIRMA.....: [ ] I
=====
I PF1 -> YARDIM
I PF3 -> ÇIKIŞ
I
=====
```

```

+-----+
I      KULLANICI KODUNUZ (USERID).....: [ ]
I      YIGIT MAKINASININ ŞİFRESİ.....: [ ]
+-----+
I      ERIŞİLECEK (ACCESS) DİSKLERİNİZİN ADRESLERİNİ GİRİNİZ
I      R/O
I      ACCESS OLUNACAK   ADR. PASSWORD   ADR. PASSWORD   ADR. PASSWORD   R/W
I      DİSKLER          [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
I      [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
I      [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
I      [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
I      [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
+-----+
I      ÇALIŞACAK OLAN   PL/1.....: [ ] COBOL.....: [ ]
I      PROGRAMLAR      FORTRAN..: [ ] WATFIV....: [ ]
I      ASSEMBLE.: [ ] VSBASIC...: [ ]
I      WBASIC.....: [ ] WFBASIC...: [ ]
I      GENSTAT...: [ ] MINTAB....: [ ]
I      SEÇENEGİNİZİN KARŞISINA "X" KOYUN
+-----+
I      PF1 -> YARDIM   PF3 -> ANA MENU
+-----+

```

```
+====+
I     ÇALIŞACAK OLAN PROGRAMIN İSMİ.....: [      ]
I====+
I     PF1 -> YARDIM      PF3 -> ANA MENU
I====+
+====+
```

```

+-----+
I PROGRAMINIZIN KULLANDIĞI KUTUKLERİN CMS'DEKİ KARŞILIKLARINI BURADAI
I BELİRTMENİZ GEREKİR. ÖRNEK;
I
I PROGRAMDA TANIMLI OLAN (DCL KUTUK FILE RECORD ...) KÜTÜK
I CMS'DEKİ KÜTÜĞÜNÜZÜN TANIMIYLA EERŞBER AŞAĞIDAKİ GİBİ TANIMINI
I GİRİN.
I
I KÜTÜK ADI .....: KÜTÜK (PROGRAMDA TANIMLI)
I CMS FN .....: CMS (CMS KÜTÜK TANIMI)
I CMS FT .....: DATA (
I CMS FM .....: A " " )
I
I BİLGİLERİNİZİ LÜTFEN AŞAĞIDAKİ FORMATTA GİRİNİZ.
I BİRDEN ÇOK KÜTÜK VE CMS KÜTÜK TANIMI İÇİN SADECE PF12 TUŞUNU KUL.
+-----+
I PROGRAMDA TANIMLI OLAN KÜTÜK ADI .....: |
I
I CMS KÜTÜK ADI (FN).....: |
I CMS KÜTÜK TİPİ (FT).....: |
I CMS KÜTÜK MODU (FM).....: |
I CMS KÜTÜK UZUNLUĞU .....: | OUTPUT....: |
+-----+
I PF1 -> YARDIM PF3 -> ANA MENU PF12 -> BİR SONR.KÜT.TAN.
I
+-----+

```

CM EXEC

DERLEME JCL'LERİNE ÖRNEK

Kullanıcı tarafından sistem çalışırken girilecek olan bilgiler daha koyu yazılmıştır.

```
CP SPOOL PUNCH TO CMSBATCH CCNT
&PUNCH /JOB CMSBATCH CENGİZ LEVENT
&PUNCH ACCESS 195 A
&PUNCH CP LINK SYSADMX 191 500 RR CENGİZ
&PUNCH ACCESS 500 B
&PUNCH ACCESS 319 P
&PUNCH ACCESS 19E Y
&PUNCH ACCESS 326 Q
&PUNCH GLOBAL MACLIB CMSLIB
&PUNCH GLOBAL TXTLIB CMSLIB PLILIB
&PUNCH PLIOPT CENGİZ
&PUNCH CP SPOOL PUNCH TO SYSADMX
&PUNCH DISK DUMP CENGİZ LISTING
&PUNCH /*
CP SPOOL PUNCH NOCONT CLOSE
&EXIT
```

ÇALIŞTIRMA JCL'LERİNE ÖRNEK

Kullanıcı tarafından sistem çalışırken girilecek olan bilgiler daha koyu yazılmıştır.

```
CP SPOOL PUNCH TO CMSBATCH CCNT
&PUNCH /JOB CMSBATCH CENGİZ LEVENT
&PUNCH ACCESS 195 A
&PUNCH CP LINK SYSADMX 191 500 RR CENGİZ
&PUNCH ACCESS 500 B
&PUNCH ACCESS 319 P
&PUNCH ACCESS 19E Y
&PUNCH ACCESS 326 Q
&PUNCH GLOBAL MACLIB CMSLIB
&PUNCH GLOBAL TXTLIB CMSLIB PLILIB
&PUNCH FILEDEF CENGİZ DISK YARDIM DATA A
      (RECFM F LRECL 70)
&PUNCH FILEDEF ÇIKTI DISK ÇIKTI DATA A
      (RECFM F LRECL 100)
&PUNCH LOAD CENGİZ
&PUNCH START
&PUNCH CP SPOOL PUNCH TO SYSADMX
&PUNCH DISK DUMP ÇIKTI DATA
&PUNCH /*
CP SPOOL PUNCH NOCONT CLOSE
&EXIT
```

BATYIL EXEC

CP TAG DEV PUNCH TRYILDIZ CMSBATCH
CP SPOOL PUNCH TO RSCS
PUNCH YILDIZ EXEC (NOH

EK-D

YILDIZ EXEC

```
/JOB CAHIT CAHIT CMSBATCH
ACCESS 319 P
ACCESS 19E Y
ACCESS 326 Q
FILEDEF INMOVE TERM (RECFM F BLOCK 80 LRECL 80
FILEDEF OUTMOVE DISK SEKIL PLIOPT A1(RECFM F LRECL 80 BLOCK 80
MOVEFILE
  SEKIL: PROCEDURE OPTIONS(MAIN);
    DCL COLON_NUM(8) CHAR(10) INIT((8)('123456789+'));
  ISLEM:
    PUT EDIT('CHAR...: C      VARY...: V      RANGE...: R')
      (COL(1),A);
    CALL NUMARA_GOS;
  NUMARA_GOS:PROC;
    PUT EDIT((COLON_NUM(I) DO I=1 TO 8))(COL(1),(8)A(10));
    DO I=2 TO 21;
      PUT EDIT(I)(COL(1),F(2));
    END;
    RETURN;
  END NUMARA_GOS;
  END SEKIL;
/*
PLIOPT SEKIL
GLOBAL TXTLIB PLILIB
FI SYSPRINT DISK SEKIL DATA A(RECFM F LRECL 80
LOAD SEKIL(START
CP TAG DEV PUNCH TREARN SYSADMS
CP SPOOL PUNCH RSCS
PUNCH SEKIL DATA A(
```


CM2 EXEC

```
/*      */
'FI HELP DISK FILE HELP A(RECFM F LRECL 75 XTENT 100'
'FI JCL DISK CM EXEC A(RECFM F LRECL 80'
"STATE" FILE USER A
IF RC = 0 THEN 'ERASE' FILE USER A1
USERID = USERID()
EXECIO 1 DISKW FILE USER A1 "(" STRING USERID
'LOAD CM2(START'
'EXEC CM'
'ERASE' FILE USER A
```

```

PROCESS MACRO,S,F,GS;
CMSBATC:PROCEDURE OPTIONS(MAIN);
%INCLUDE SYSLIB(EUDPLI);
DCL
/*****
Yığıt makinasının kullanılabilmesi için programın yarattığı
iş kontrol kartlarının yazıldığı kütük tanım
*****/
      JCL FILE RECORD OUTPUT SEQL ENV(F RECSIZE(80)),
      USER FILE RECORD INPUT SEQL ENV(V RECSIZE(80)),

/*****
iş kontrol kart kütüğü için gerekli olan yapı
*****/
      JCL_CARD CHAR(80),

/*****
Programın çalışması sırasında üreteceği kütüklerin isimleri, tipleri
modelleri ve record uzunlukları
*****/
      OUTPUT_NAME(20) CHAR(8) ,
      OUTPUT_TYPE(20) CHAR(8) ,
      OUTPUT_MODE(20) CHAR(1) ,
      OUTPUT_BLOCK(20) CHAR(6) ,
      STR CHAR(75) INIT(' ');
      DO IJ = 1 TO 20;
          OUTPUT_NAME(IJ) = '          ';OUTPUT_TYPE(IJ) = '          ';
          OUTPUT_MODE(IJ) = ' ';          OUTPUT_BLOCK(IJ) = '          ';
      END;

/*****
Ekrandan girilen alanların boş olup olmadığının kontrolü için
gerekli olan değişken tanımı
*****/
DCL
      BOS_KONTROL CHAR(54)
      INIT('0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ@#%&*()_+=-:?.,\/' ),

/*****
Kullanılan değişken tanımları
*****/
      (TANIM,KLM,III,KKK,KUT,MM,I,K) FIXED BIN,

/*****
Kullanıcının bağlanacağı diskleri belirten değişken
*****/
      HARF(21) CHAR(1) INIT('B','C','D','E','F','G',
                          'H','I','J','K','L','M',
                          'N','O','R','T','U','V','W','X','Z'),

```

```

/*****
Ekranada kullanılan deęişkenler için gerekli tanımlamalar
*****/
    STRING_BATCH01    CHAR(191) DEF BATCH01_YAPI,
    STRING_BATCH03    CHAR(31)   DEF BATCH03_YAPI,

/*****
Kullanıcının verdiği disk ve bunu yazma yada okuma şifrelerinin
bulunduęu deęişken
*****/
    DISK_PASS        CHAR(165) INIT(' '),

/*****
Kullanıcının derleyeceęi yada çalıştıracak programın tipine
göre kullanılacak kütüphaneleri belirleyen deęişken
*****/
    KUTUPHANE        CHAR(10) INIT (' '),

/*****
Disklerin okuma şifrelerini içeren deęişken
*****/
    READ_PASS        CHAR(8) INIT(' '),

/*****
Disklerin bağlanacak varsayılan adresleri içeren deęişken
*****/
    DISK_VIRT        PIC'999' INIT(0),

/*****
Kullanıcının program çalışırken yada derlenirken
baęlanacak kendi disklerinin adresleri
ayrıca kullanıcının kullanım kodunu belirleyen deęişken
*****/
    USERID_VM        CHAR(80) INIT(' '),
    USERDISK          CHAR(3)  INIT(' '),

/*****
Okuma-yazma durumunda bağlanılacak olan disklerde okuma yazma
yı belirten deęişken
*****/
    READWRITE        CHAR(5) INIT(' '),

/*****
Program derlenecekse hangi derleyicinin kullanılacağını
gösteren deęişken
*****/
    COMPILER          CHAR(8)  INIT(' '),

/*****
Program çalışacaksa nasıl çalıştırılacağını gösteren deęişken
*****/
    LOADER            CHAR(8)  INIT(' '),

```

```

/*****
Program çalışırken hangi macro kütüphanelerine bağlanılacağını
gösteren değişken
*****/
MACLIB          CHAR(60) VAR INIT(' '),

/*****
Program çalışırken hangi text kütüphanelerine bağlanacağını
gösteren değişken
*****/
TXTLIB          CHAR(60) VAR INIT(' '),

/*****
Programda kullanılan PF tuşlarını kontrol eden değişken
*****/
STATUS BIT(8)  INIT((8)'0'B),

/*****
Program çalışırken kullanıcıya gerekli mesajların ekranda
gösterilmesini sağlayan değişkenler
*****/
COMMENT CHAR(80),

/*****
Help kullanıldığı zaman hangi menüden help istendiğini
gösteren değişken
*****/
LEVEL    FIXED BIN;

/*****
*****   Derleme ve Yükleme için Yapı tanımı   *****
*****/
DCL 1 BATCH00,
    2 ADRES(2) PTR INIT(ADDR(DERLE),ADDR(CALIS)),
    2 SON00 BIT(32) INIT((32)'1'B);
DCL 1 BATCH00_YAPI,
    2 DERLE    CHAR(1)  INIT(' '),
    2 CALIS    CHAR(1)  INIT(' ');

```

```

/*****
***** - Kullanıcı adı ve yiğit makinasının şifresi *****
***** - Yazma ve okuma durumunda bağlanılacak *****
***** disklerin adresleri ve şifreleri *****
***** - Kullanılacak programlama dili için *****
***** gerekli kütüphanelerin belirtimi için *****
***** yapı tanımı *****
*****/

```

```
DCL 1 BATCH01,
```

```

  2 ADRES(42) PTR INIT(ADDR(USERID),ADDR(PASSWORD),
    ADDR(ACCESS1),ADDR(PASSWORD1),
    ADDR(ACCESS2),ADDR(PASSWORD2),
    ADDR(ACCESW1),ADDR(PASSWORW1),
    ADDR(ACCESS3),ADDR(PASSWORD3),
    ADDR(ACCESS4),ADDR(PASSWORD4),
    ADDR(ACCESW2),ADDR(PASSWORW2),
    ADDR(ACCESS5),ADDR(PASSWORD5),
    ADDR(ACCESS6),ADDR(PASSWORD6),
    ADDR(ACCESW3),ADDR(PASSWORW3),
    ADDR(ACCESS7),ADDR(PASSWORD7),
    ADDR(ACCESS8),ADDR(PASSWORD8),
    ADDR(ACCESW4),ADDR(PASSWORW4),
    ADDR(ACCESS9),ADDR(PASSWORD9),
    ADDR(ACCESS10),ADDR(PASSWORD10),
    ADDR(ACCESW5),ADDR(PASSWORW5),
    ADDR(LIBRARY1),ADDR(LIBRARY2),
    ADDR(LIBRARY3),ADDR(LIBRARY4),
    ADDR(LIBRARY5),ADDR(LIBRARY6),
    ADDR(LIBRARY7),ADDR(LIBRARY8),
    ADDR(LIBRARY9),ADDR(LIBRARY10)),
  2 SON01 BIT(32) INIT((32)'1'B);

```

```
DCL 1 BATCH01_YAPI,
```

```

  2 USERID CHAR(8) INIT(' '),2 PASSWORD CHAR(8) INIT(' '),
  2 ACCESS1 CHAR(3) INIT(' '),2 PASSWORD1 CHAR(8) INIT(' '),
  2 ACCESS2 CHAR(3) INIT(' '),2 PASSWORD2 CHAR(8) INIT(' '),
  2 ACCESW1 CHAR(3) INIT(' '),2 PASSWORW1 CHAR(8) INIT(' '),
  2 ACCESS3 CHAR(3) INIT(' '),2 PASSWORD3 CHAR(8) INIT(' '),
  2 ACCESS4 CHAR(3) INIT(' '),2 PASSWORD4 CHAR(8) INIT(' '),
  2 ACCESW2 CHAR(3) INIT(' '),2 PASSWORW2 CHAR(8) INIT(' '),
  2 ACCESS5 CHAR(3) INIT(' '),2 PASSWORD5 CHAR(8) INIT(' '),
  2 ACCESS6 CHAR(3) INIT(' '),2 PASSWORD6 CHAR(8) INIT(' '),
  2 ACCESW3 CHAR(3) INIT(' '),2 PASSWORW3 CHAR(8) INIT(' '),
  2 ACCESS7 CHAR(3) INIT(' '),2 PASSWORD7 CHAR(8) INIT(' '),
  2 ACCESS8 CHAR(3) INIT(' '),2 PASSWORD8 CHAR(8) INIT(' '),
  2 ACCESW4 CHAR(3) INIT(' '),2 PASSWORW4 CHAR(8) INIT(' '),
  2 ACCESS9 CHAR(3) INIT(' '),2 PASSWORD9 CHAR(8) INIT(' '),
  2 ACCESS10 CHAR(3) INIT(' '),2 PASSWORD10 CHAR(8) INIT(' '),
  2 ACCESW5 CHAR(3) INIT(' '),2 PASSWORW5 CHAR(8) INIT(' '),
  2 LIBRARY1 CHAR(1) INIT(' '),2 LIBRARY2 CHAR(1) INIT(' '),
  2 LIBRARY3 CHAR(1) INIT(' '),2 LIBRARY4 CHAR(1) INIT(' '),
  2 LIBRARY5 CHAR(1) INIT(' '),2 LIBRARY6 CHAR(1) INIT(' '),
  2 LIBRARY7 CHAR(1) INIT(' '),2 LIBRARY8 CHAR(1) INIT(' '),
  2 LIBRARY9 CHAR(1) INIT(' '),2 LIBRARY10 CHAR(1) INIT(' ');

```

EK-G4

```

/*****
*****      Derlenecek program ismi için      *****/
*****      gerekli yapı                      *****/
*****/
DCL 1 BATCH02,
    2 ADRES(1) PTR INIT(ADDR(PRGMID)),
    2 SON02 BIT(32) INIT((32)'1'B);
DCL 1 BATCH02_YAPI,
    2 PRGMID CHAR(8) INIT(' ');

/*****
*****      Programda kullanılan kütüklerin tanımları *****/
*****      için gerekli yapı                      *****/
*****/
DCL 1 BATCH03,
    2 ADRES(6) PTR INIT(ADDR(FILENAME),ADDR(CMSNAME),
                        ADDR(CMSTYPE),ADDR(CMSMODE),
                        ADDR(CMSBLOCK),ADDR(OUTPUT)),
    2 SON03 BIT(32) INIT((32)'1'B);
DCL 1 BATCH03_YAPI,
    2 FILENAME CHAR(8) INIT(' '),2 CMSNAME CHAR(8) INIT(' '),
    2 CMSTYPE CHAR(8) INIT(' '),2 CMSMODE CHAR(1) INIT(' '),
    2 CMSBLOCK CHAR(6) INIT(' '),2 OUTPUT CHAR(1) INIT(' ');

/*****
*****      Çalışacak program ismi için      *****/
*****      gerekli yapı                      *****/
*****/
DCL 1 BATCH04,
    2 ADRES(1) PTR INIT(ADDR(PRGMID1)),
    2 SON04 BIT(32) INIT((32)'1'B);
DCL 1 BATCH04_YAPI,
    2 PRGMID1 CHAR(8) INIT(' ');

OPEN FILE(JCL) OUTPUT;
READ FILE(USER) INTO(USERID_VM);

DISPLAY2:
COMMENT = ' ';
STRING(BATCH00_YAPI)=LOW(LENGTH(STRING(BATCH00_YAPI)));
PDISPLAY PNAME('BATCH0 ') RSTATUS(STATUS)
LDLIST(BATCH00) ULDLIST(BATCH00) COMMENT(COMMENT);
LEVEL = 1;
IF STATUS = PF1 THEN DO;CALL HELP1(LEVEL);GOTO DISPLAY2;END;
IF STATUS = PF3 THEN DO;CLOSE FILE(JCL);STOP;END;
MM = INDEX(BOS_KONTROL,DERLE);
IF MM ^= 0 THEN DO;CALL DERLEME;GOTO DISPLAY2;END;
MM = INDEX(BOS_KONTROL,CALIS);
IF MM ^= 0 THEN DO;CALL CALISMA;GOTO DISPLAY2;END;
ELSE GOTO DISPLAY2;

I = 0;

```



```

/*****
*****      Program çalıştırma ile ilgili olan Kontrol      *****
*****      kartlarını oluşturan altyordam                  *****
*****/
CALISMA:PROC;
  COMMENT = ' ';I = 0;
DISPLAY:
  STRING(BATCH01_YAPI)=LOW(LENGTH(STRING(BATCH01_YAPI)));
  STRING(BATCH02_YAPI)=LOW(LENGTH(STRING(BATCH02_YAPI)));
  STRING(BATCH03_YAPI)=LOW(LENGTH(STRING(BATCH03_YAPI)));
DISPLAY5:
  PDISPLAY PNAME('BATCH1 ') RSTATUS(STATUS)
  LDLIST(BATCH01) ULDLIST(BATCH01) COMMENT(COMMENT);
  LEVEL = 2;
  IF STATUS = PF1 THEN DO;CALL HELP1(LEVEL);GOTO DISPLAY;END;
  IF STATUS = PF3 THEN DO COMMENT = ' ';RETURN;END;
  IF USERID ^= USERID_VM THEN DO;
    COMMENT = 'KULLANICI KODUNUZU YANLIS GIRDINIZ BIRDAHA GIRIN';
    GOTO DISPLAY5;

    END;
  IF PASSWORD ^= 'CENGIZ ' THEN DO;
    COMMENT='SIFRENIZ YANLIS LUTFEN BIR DAHA GIRIN';
    I = I + 1;
    IF I > 2 THEN CALL BIT;
    GOTO DISPLAY5;

    END;

  KKK = 0;
  KUTUPHANE = SUBSTR(STRING_BATCH01,182,10);
  DO III = 1 TO 10;
    KUT = INDEX(BOS_KONTROL,SUBSTR(KUTUPHANE,III,1));
    IF KUT = 0 THEN KKK=KKK+1;
  END;
  IF KKK = 10 THEN DO;
    COMMENT = 'KUTUPHANE VEYA DISK SECIMI YAPMADINIZ'
    GOTO DISPLAY5;
  END;
  JCL_CARD='CP SPOOL PUNCH TO CMSBATCH CONT';
  WRITE FILE(JCL) FROM(JCL_CARD);
  JCL_CARD='&PUNCH /JOB CMSBATCH CENGIZ ']]'LEVENT ' ';
  WRITE FILE(JCL) FROM(JCL_CARD);
  JCL_CARD='&PUNCH ACCESS 195 A';
  WRITE FILE(JCL) FROM(JCL_CARD);
  DISK_PASS = SUBSTR(STRING_BATCH01,17,165);
  DISK_VIRT = 500;K=1;JJ=0;
  DO I = 1 TO 165 BY 11;
    JJ=JJ+1 ; /* R/W ACCESS ICIN SAYAC */
    USERDISK = SUBSTR(DISK_PASS,I,3);
    READ_PASS= SUBSTR(DISK_PASS,I+3,8);
    MM = INDEX(BOS_KONTROL,SUBSTR(USERDISK,1,1));
  IF JJ = 3 THEN DO;
    JJ = 0;
    READWRITE = ' MR ';
    END;
  ELSE READWRITE = ' RR ';

```

EK-G6

```

IF MM ^= 0 THEN DO;
    JCL_CARD='&PUNCH CP LINK ']]USERID]]' ']]
    USERDISK]]' ']] DISK_VIRT ]]READWRITE]]READ_PASS;
    WRITE FILE(JCL) FROM(JCL_CARD);
    JCL_CARD='&PUNCH ACCESS ']] DISK_VIRT ]]' ']]HARF(K);
    WRITE FILE(JCL) FROM(JCL_CARD);
        DISK_VIRT = DISK_VIRT + 1;
        K=K+1;

    END;

END;
JCL_CARD='&PUNCH ACCESS 319 P';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH ACCESS 19E Y';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH ACCESS 326 Q';
WRITE FILE(JCL) FROM(JCL_CARD);
MACLIB = ' ';TXTLIB = ' ';
MM = INDEX(BOS_KONTROL,LIBRARY1);
IF MM ^= 0 THEN DO;TXTLIB = TXTLIB]]'PLILIB ' ;
    LOADER = ' LOAD ' ;END;
MM = INDEX(BOS_KONTROL,LIBRARY2);
IF MM ^= 0 THEN DO;
    TXTLIB = TXTLIB]]'COBOLVS COBLIBVS ' ;
    MACLIB = MACLIB]]'COBOLVS ' ;
    LOADER = ' LOAD ' ;
    END;
MM = INDEX(BOS_KONTROL,LIBRARY3);
IF MM ^= 0 THEN DO;
    TXTLIB=TXTLIB]]'IFXOTRCE VALTLIB VFORTLIB NAG1LIB NAG2LIB ' ;
    LOADER = ' LOAD ' ;END;
MM = INDEX(BOS_KONTROL,LIBRARY4);
IF MM ^= 0 THEN LOADER = 'WATFIV ' ;
MM = INDEX(BOS_KONTROL,LIBRARY5);
IF MM ^= 0 THEN DO;
    MACLIB=MACLIB]]'DMSSP OSMACRO1 OSMACRO TSOMAC ' ;
    LOADER = 'LOAD ' ;
    END;
MM = INDEX(BOS_KONTROL,LIBRARY6);
IF MM ^= 0 THEN LOADER = 'VSBASIC ' ;
MM = INDEX(BOS_KONTROL,LIBRARY7);
IF MM ^= 0 THEN LOADER = 'WBASIC ' ;
MM = INDEX(BOS_KONTROL,LIBRARY8);
IF MM ^= 0 THEN LOADER = 'WPASCAL ' ;
JCL_CARD='&PUNCH GLOBAL MACLIB CMSLIB']]MACLIB;
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH GLOBAL TXTLIB CMSLIB']]TXTLIB;
WRITE FILE(JCL) FROM(JCL_CARD);
COMMENT = ' ';

```



```

DISPLAY15:
  PDISPLAY PNAME('BATCH2 ') RSTATUS(STATUS)
  LDLIST(BATCH02) ULDLIST(BATCH02) COMMENT(COMMENT);
  LEVEL = 3;
  IF STATUS = PF1 THEN DO;CALL HELP1(LEVEL);GOTO DISPLAY15;END;
  IF STATUS = PF3 THEN DO;COMMENT = ' ';RETURN;END;
  MM = INDEX(BOS_KONTROL,SUBSTR(PRGMID,1,1));
  IF MM = 0 THEN DO;
    COMMENT = 'PROGRAM ISMI GIRMEK ZORUNDASINIZ';
    GOTO DISPLAY15;
  END;

TANIM = 1;

BATCH33:
  PDISPLAY PNAME('BATCH3 ') RSTATUS(STATUS)
  LDLIST(BATCH03) ULDLIST(BATCH03) COMMENT(COMMENT);
  LEVEL = 4;
  IF STATUS = PF1 THEN DO;CALL HELP1(LEVEL);GOTO BATCH33;END;
  IF STATUS = PF3 THEN DO;COMMENT = ' ';RETURN;END;
  MM = INDEX(BOS_KONTROL,SUBSTR(STRING_BATCH03,1,1));
  IF MM = 0 THEN DO;
    COMMENT = 'PROGRAMDA KULLANILAN ISMI BOS GECEMEZSINIZ';
    GOTO BATCH33;
  END;
  MM = INDEX(BOS_KONTROL,SUBSTR(STRING_BATCH03,9,1));
  IF MM = 0 THEN DO;
    COMMENT = 'CMS DISK ISMINI BOS GECEMEZSINIZ';
    GOTO BATCH33;
  END;
  MM = INDEX(BOS_KONTROL,SUBSTR(STRING_BATCH03,17,1));
  IF MM = 0 THEN DO;
    COMMENT = 'CMS DISK TIPINI BOS GECEMEZSINIZ';
    GOTO BATCH33;
  END;
  MM = INDEX(BOS_KONTROL,SUBSTR(STRING_BATCH03,25,1));
  IF MM = 0 THEN DO;
    COMMENT = 'CMS DISK MODUNU BOS GECEMEZSINIZ';
    GOTO BATCH33;
  END;
  MM = INDEX(BOS_KONTROL,SUBSTR(STRING_BATCH03,26,1));
  IF MM = 0 THEN DO;
    COMMENT = 'CMS DISK UZUNLUGUNU BOS GECEMEZSINIZ';
    GOTO BATCH33;
  END;
  IF STATUS = PF12 THEN DO;
    MM = INDEX(BOS_KONTROL,OUTPUT);
    IF MM = 0 THEN DO;
      OUTPUT_NAME(TANIM) = CMSNAME;
      OUTPUT_TYPE(TANIM) = CMSTYPE;
      OUTPUT_BLOCK(TANIM) = CMSBLOCK;
      OUTPUT_MODE(TANIM) = CMSMODE;
      TANIM = TANIM + 1;
    END;
  END;

```

```

JCL_CARD='&PUNCH FILEDEF ']]FILENAME]]' DISK ']]
          CMSNAME]]' ']]CMSTYPE]]' ']]CMSMODE]]
          '(RECFM F LRECL ']]CMSBLOCK;
WRITE FILE(JCL) FROM(JCL_CARD);
          JCL_CARD = ' ';
STRING(BATCH03_YAPI)=LOW(LENGTH(STRING(BATCH03_YAPI)));
          GOTO BATCH33;
          END;
MM = INDEX(BOS_KONTROL,OUTPUT);
IF MM ^= 0 THEN DO;
    OUTPUT_NAME(TANIM) = CMSNAME;
    OUTPUT_TYPE(TANIM) = CMSTYPE;
    OUTPUT_BLOCK(TANIM) = CMSBLOCK;
    OUTPUT_MODE(TANIM) = CMSMODE;
    END;
JCL_CARD='&PUNCH FILEDEF ']]FILENAME]]' DISK ']]
          CMSNAME]]' ']]CMSTYPE]]' ']]CMSMODE]]
          '(RECFM F LRECL ']]CMSBLOCK;
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH ']]LOADER]]' ']]PRGMID ;
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH START';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH CP SPOOL PUNCH TO ']]USERID;
WRITE FILE(JCL) FROM(JCL_CARD);
DO KLM = 1 TO TANIM;
    JCL_CARD='&PUNCH DISK DUMP ']]OUTPUT_NAME(KLM)]]' '
    ]]OUTPUT_TYPE(KLM);
    WRITE FILE(JCL) FROM(JCL_CARD);
END;

JCL_CARD='&PUNCH /*';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='CP SPOOL PUNCH NOCONT CLOSE';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&EXIT';
WRITE FILE(JCL) FROM(JCL_CARD);

RETURN;
END CALISMA;

```

```

/*****
*****      Program derleme      ile ilgili olan Kontrol      *****/
*****      kartlarını oluşturan altyordam      *****/
*****/
DERLEME:PROC;
    COMMENT = ' ';I = 0;
DISPLAY1:
    STRING(BATCH01_YAPI)=LOW(LENGTH(STRING(BATCH01_YAPI)));
    STRING(BATCH02_YAPI)=LOW(LENGTH(STRING(BATCH02_YAPI)));
    STRING(BATCH03_YAPI)=LOW(LENGTH(STRING(BATCH03_YAPI)));
DISPLAY11:
    PDISPLAY PNAME('BATCH1 ') RSTATUS(STATUS)
    LDLIST(BATCH01) ULDLIST(BATCH01) COMMENT(COMMENT);
    LEVEL = 5;

```

EK-G9

```

IF STATUS = PF1 THEN DO;CALL HELP1(LEVEL);GOTO DISPLAY1;END;
IF STATUS = PF3 THEN DO;COMMENT = ' ';RETURN;END;
IF USERID ^= USERID_VM THEN DO;
  COMMENT = 'KULLANICI KODUNUZU YANLIS GIRDINIZ BIRDAHA GIRIN';
  GOTO DISPLAY11;
  END;
IF PASSWORD ^= 'CENGIZ ' THEN DO;
  COMMENT='SIFRENIZ YANLIS LUTFEN BIR DAHA GIRIN';
  I = I + 1;
  IF I > 2 THEN CALL BIT;
  GOTO DISPLAY11;
  END;
KKK = 0;
KUTUPHANE = SUBSTR(STRING_BATCH01,182,10) ;
DO III = 1 TO 10;
  KUT = INDEX(BOS_KONTROL,SUBSTR(KUTUPHANE,III,1));
  IF KUT = 0 THEN KKK=KKK+1;
  END;
IF KKK = 10 THEN DO;
  COMMENT = 'KUTUPHANE VEYA DISK SECIMI YAPMADINIZ'
  GOTO DISPLAY11;
  END;
JCL_CARD='CP SPOOL PUNCH TO CMSBATCH CONT';
  WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH /JOB CMSBATCH CENGIZ ']]'LEVENT '];
  WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH ACCESS 195 A';
  WRITE FILE(JCL) FROM(JCL_CARD);
DISK_PASS = SUBSTR(STRING_BATCH01,17,165);
DISK_VIRT = 500;K=1;JJ=0;
DO I = 1 TO 165 BY 11;
  JJ=JJ+1 ; /* R/W ACCESS ICIN SAYAC */
  USERDISK = SUBSTR(DISK_PASS,I,3);
  READ_PASS= SUBSTR(DISK_PASS,I+3,8);
  MM = INDEX(BOS_KONTROL,SUBSTR(USERDISK,1,1));
IF JJ = 3 THEN DO;
  JJ = 0;
  READWRITE = ' MR ';
  END;
  ELSE READWRITE = ' RR ';
IF MM ^= 0 THEN DO;
  JCL_CARD='&PUNCH CP LINK ']]USERID]]' ']]
  USERDISK]]' ']] DISK_VIRT ]]]READWRITE]]READ_PASS;
  WRITE FILE(JCL) FROM(JCL_CARD);
  JCL_CARD='&PUNCH ACCESS ']] DISK_VIRT ]]]' ']]HARF(K);
  WRITE FILE(JCL) FROM(JCL_CARD);
  DISK_VIRT = DISK_VIRT + 1;
  K=K+1;
  END;
END;
END;

```

```

JCL_CARD='&PUNCH ACCESS 319 P';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH ACCESS 19E Y';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH ACCESS 326 Q';
WRITE FILE(JCL) FROM(JCL_CARD);
MACLIB = ' ';TXTLIB = ' ';
MM = INDEX(BOS_KONTROL,LIBRARY1);
IF MM ^= 0 THEN DO;TXTLIB = TXTLIB]]'PLILIB ' ;
COMPILER = ' PLIOPT ' ;END;
MM = INDEX(BOS_KONTROL,LIBRARY2);
IF MM ^= 0 THEN DO;
TXTLIB = TXTLIB]]'COBOLVS COBLIBVS ' ;
MACLIB = MACLIB]]'COBOLVS ' ;
COMPILER = ' COBOL ' ;
END;
MM = INDEX(BOS_KONTROL,LIBRARY3);
IF MM ^= 0 THEN DO;
TXTLIB=TXTLIB]]'IFXOTRCE VALTLIB VFORTLIB NAG1LIB NAG2LIB ' ;
COMPILER = ' FORTVS ' ;END;
MM = INDEX(BOS_KONTROL,LIBRARY4);
IF MM ^= 0 THEN COMPILER = 'WATFIV ' ;
MM = INDEX(BOS_KONTROL,LIBRARY5);
IF MM ^= 0 THEN DO;
MACLIB=MACLIB]]'DMSSP OSMACRO1 OSMACRO TSOMAC ' ;
COMPILER = 'ASSEMBLE' ;
END;
MM = INDEX(BOS_KONTROL,LIBRARY6);
IF MM ^= 0 THEN COMPILER = 'VSBASIC ' ;
MM = INDEX(BOS_KONTROL,LIBRARY7);
IF MM ^= 0 THEN COMPILER = 'WBASIC ' ;
MM = INDEX(BOS_KONTROL,LIBRARY8);
IF MM ^= 0 THEN COMPILER = 'WPASCAL ' ;
JCL_CARD='&PUNCH GLOBAL MACLIB CMSLIB']]MACLIB;
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH GLOBAL TXTLIB CMSLIB']]TXTLIB;
WRITE FILE(JCL) FROM(JCL_CARD);
COMMENT = ' ';
DISPLAY3:
PDISPLAY PNAME('BATCH4 ') RSTATUS(STATUS)
LDLIST(BATCH04) ULDLIST(BATCH04) COMMENT(COMMENT);
LEVEL = 6;
IF STATUS = PF1 THEN DO;CALL HELP1(LEVEL);GOTO DISPLAY3;END;
IF STATUS = PF3 THEN DO;COMMENT = ' ';RETURN;END;
MM = INDEX(BOS_KONTROL,SUBSTR(PRGMID1,1,1));
IF MM = 0 THEN DO;
COMMENT = 'PROGRAM ISMI GIRMEK ZORUNDASINIZ';
GOTO DISPLAY3;
END;

```

```

JCL_CARD='&PUNCH ']]COMPILER]]' ']]PRGMID1;
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH CP SPOOL PUNCH TO ']]USERID;
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH DISK DUMP ']]PRGMID1]]' LISTING ';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&PUNCH /*';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='CP SPOOL PUNCH NOCONT CLOSE';
WRITE FILE(JCL) FROM(JCL_CARD);
JCL_CARD='&EXIT';
WRITE FILE(JCL) FROM(JCL_CARD);

```

```

RETURN;
END DERLEME;

```

```

/*****
*****      Help altyordami      *****/
*****/
HELP1:PROCEDURE(SEVIYE);
DCL (BOSSAY,BOSYER,BOSLUK_SAYISI,DUMMY_INDIS,SEVIYE,INDIS) FIXED BIN;
DCL (JJ,IJ,J) FIXED BIN;
DCL HELP FILE RECORD DIRECT ENV(REGIONAL(1) F RECSIZE(75));
DCL HELP_STRUCTURE CHAR(75) INIT(' ');
DCL HELP_STRUCTURE_ARRAY(100) CHAR(75);
DO IJ=1 TO 100;HELP_STRUCTURE_ARRAY(IJ)=STR;END;
DCL HELPKEY PIC'99999';
DCL (BAS,I) FIXED BIN;
DCL HELP_YAPI_STRUCTURE CHAR(1500) DEF HELP_YAPI;
DCL 1 HELPSCRN,
    2 ADRES(20) PTR INIT(ADDR(HELP1),ADDR(HELP2),
                        ADDR(HELP3),ADDR(HELP4),
                        ADDR(HELP5),ADDR(HELP6),
                        ADDR(HELP7),ADDR(HELP8),
                        ADDR(HELP9),ADDR(HELP10),
                        ADDR(HELP11),ADDR(HELP12),
                        ADDR(HELP13),ADDR(HELP14),
                        ADDR(HELP15),ADDR(HELP16),
                        ADDR(HELP17),ADDR(HELP19),
                        ADDR(HELP19),ADDR(HELP20)),
    2 HELP_SON BIT(32) INIT((32)'1'B);
DCL 1 HELP_YAPI,
    2 HELP1 CHAR(75) INIT(' '), 2 HELP2 CHAR(75) INIT(' '),
    2 HELP3 CHAR(75) INIT(' '), 2 HELP4 CHAR(75) INIT(' '),
    2 HELP5 CHAR(75) INIT(' '), 2 HELP6 CHAR(75) INIT(' '),
    2 HELP7 CHAR(75) INIT(' '), 2 HELP8 CHAR(75) INIT(' '),
    2 HELP9 CHAR(75) INIT(' '), 2 HELP10 CHAR(75) INIT(' '),
    2 HELP11 CHAR(75) INIT(' '), 2 HELP12 CHAR(75) INIT(' '),
    2 HELP13 CHAR(75) INIT(' '), 2 HELP14 CHAR(75) INIT(' '),
    2 HELP15 CHAR(75) INIT(' '), 2 HELP16 CHAR(75) INIT(' '),
    2 HELP17 CHAR(75) INIT(' '), 2 HELP18 CHAR(75) INIT(' '),
    2 HELP19 CHAR(75) INIT(' '), 2 HELP20 CHAR(75) INIT(' ');

```

```

DCL COMMENT CHAR(80) INIT(' ');
DCL STATUS BIT(8) INIT((8)'0'B);
DCL YAZIM CHAR(75) INIT(' ');
DCL TAMPON CHAR(75) INIT(' ') VAR;
DCL (RETURN_STATUS,OKUNAN) FIXED BIN;
OPEN FILE(HELP) INPUT;
STRING(HELP_YAPI)=LOW(LENGTH(STRING(HELP_YAPI)));
SELECT;
  WHEN (SEVIYE = 1) DO;
    HELPKEY=0;INDIS = 1;CALL EKRAM_HELP;RETURN;
    END;
  WHEN (SEVIYE = 2) DO;
    HELPKEY=17;INDIS=1;CALL EKRAM_HELP;RETURN;
    END;
  WHEN (SEVIYE = 3) DO;
    HELPKEY=55;INDIS=1;CALL EKRAM_HELP;RETURN;
    END;
  WHEN (SEVIYE = 4) DO;
    HELPKEY=59;INDIS=1;CALL EKRAM_HELP;RETURN;
    END;
  WHEN (SEVIYE = 5) DO;
    HELPKEY=38;INDIS=1;CALL EKRAM_HELP;RETURN;
    END;
  WHEN (SEVIYE = 6) DO;
    HELPKEY=34;INDIS=1;CALL EKRAM_HELP;RETURN;
    END;
  OTHERWISE;
END;

/*****
*****      Ekrama help menüsü yazan bölüm      *****/
*****/
DISPLAY:PROC;
  PDISPLAY PNAME('HELP ') RSTATUS(STATUS)
  LDLIST(HELPSCRN) ULDLIST(HELPSCRN) COMMENT(COMMENT);
SELECT;
  WHEN(STATUS = PF3) DO;RETURN_STATUS = 3;RETURN;END;
  WHEN(STATUS = PF7) DO;RETURN_STATUS = 7;RETURN;END;
  WHEN(STATUS = PF8) DO;RETURN_STATUS = 8;RETURN;END;
  OTHERWISE DO;RETURN_STATUS = 0;RETURN;END;
END;
END DISPLAY;

/*****
*****      Ekrama help menüsü ile ilgili bölümü yazıldığı      *****/
*****      sırada sağdan gerekli boşlukları atayan bölüm      *****/
*****/
BOS_EKLE:PROC;
DCL (ILKBOS,BOSEKLE,SONHARF) FIXED BIN;
DCL (SONBOLUM,ILKBOLUM) CHAR(75) VAR;
SONHARF = 75;BOSSAY = 0;

```



```

DO UNTIL (SUBSTR(HELP_STRUCTURE,SONHARF,1) ^= ' ');
  SONHARF = SONHARF - 1;
END;
DO I=1 TO SONHARF;
  IF SUBSTR(HELP_STRUCTURE,I,1)=' ' THEN BOSSAY=BOSSAY+1;
END;
BOSEKLE = 75 - SONHARF;
IF BOSSAY < 5 THEN RETURN;
ILKBOS=1;BOSYER=0;
DO UNTIL (BOSEKLE <= 0);
  DO UNTIL(SUBSTR(HELP_STRUCTURE,ILKBOS,1) = ' ');
    ILKBOS = ILKBOS + 1;
  END;
  DO UNTIL(SUBSTR(HELP_STRUCTURE,ILKBOS,1) ^= ' ');
    ILKBOS = ILKBOS + 1;
  END;
  ILKBOS = ILKBOS - 1;
  BOSYER = BOSYER + 1;
  ILKBOLUM = SUBSTR(HELP_STRUCTURE,1,ILKBOS);
  SONBOLUM = SUBSTR(HELP_STRUCTURE,ILKBOS+1);
  HELP_STRUCTURE = ILKBOLUM]]' ']]SONBOLUM;
  BOSEKLE = BOSEKLE - 1;
  ILKBOS = ILKBOS + 2;
  IF BOSYER > BOSSAY THEN DO;BOSYER = 0;ILKBOS = 1;END;
END;
END BOS_EKLE;

```

```

/*****
*****      Kullanıcı help istediği zaman istediği bölümün      ****
*****      ekrana yazmak için gerekli değişkenleri set eden    ****
*****      bölüm                                                ****
*****/
EKRAN_YAZMAK:PROC;
DCL DEVAM FIXED BIN;
HELP_YAPI_STRUCTURE = ' ';
DEVAM = 1;
BAS = 1;
DO UNTIL (DEVAM > 20);
  IF HELP_STRUCTURE_ARRAY(J) = '*'
    THEN DO;
    COMMENT='                                HELP MENUSU SONU';
    RETURN;
  END;
  SUBSTR(HELP_YAPI_STRUCTURE,BAS,75)=HELP_STRUCTURE_ARRAY(J);
  DEVAM = DEVAM+1;
  BAS=BAS+75;
  J=J+1;
END;
END EKRAN_YAZMAK;

```

```

/*****
*****      Kullanıcının help istediği zaman ekrana yazımını      ****
*****      sağlayan altyordam                                     ****
*****/
EKRAN_HELP:PROC;
  READ FILE(HELP) INTO(HELP_STRUCTURE) KEY(HELPKEY);
  DO UNTIL(SUBSTR(HELP_STRUCTURE,1,2) = '**');
    CALL BOS_EKLE;
    HELP_STRUCTURE_ARRAY(INDIS) = HELP_STRUCTURE;
    INDIS = INDIS + 1;
    HELPKEY = HELPKEY + 1;
    READ FILE(HELP) INTO(HELP_STRUCTURE) KEY(HELPKEY);
  END;
  INDIS = INDIS + 1;
  HELP_STRUCTURE_ARRAY(INDIS) = '**';
I_BIR:      I = 1;J=1;
I_FARKLI:   CALL EKRAN_YAZMAK;
DISPLAY_BOS:CALL DISPLAY;
  IF RETURN_STATUS = 0 THEN GOTO DISPLAY_BOS;
  IF RETURN_STATUS = 3 THEN RETURN;
  IF RETURN_STATUS = 7 THEN DO;
    COMMENT = ' ' ;
    IF (I-20) <= 20 THEN GOTO I_BIR;
    ELSE DO;I=I-20;J=I;
    GOTO I_FARKLI;END;
  END;
  IF RETURN_STATUS = 8 THEN DO;
    IF ((I+20)-INDIS) < 20 THEN DO;
      I = I + 20;J=I;
      GOTO I_FARKLI;
    END;
    ELSE IF (I+20) > INDIS-20 THEN
      GOTO DISPLAY_BOS;
    ELSE
      DO;I=I+20;J=I;GOTO I_FARKLI;END;
  END;
END EKRAN_HELP;
END HELP1;

```



```
/******  
*****   Programda şifre sözcüğü üç defada da bilinmediği   ****  
*****   takdirde gerekli olan mesajın gösterilip           ****  
*****   programın kesildiği bölüm                          ****  
*****/
```

```
BIT:PROC;  
DCL PLI3270           EXTERNAL ENTRY;  
CALL PLI3270;  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY(' ');  
DISPLAY('   YETKILI KISI ILE GORUSUN           TELEFON (244)');  
STOP;  
END BIT;  
END CMSBATC;
```