

Tamponlama (Buffering)

Bir Byte'ın İzlediği Yol

Bir programda `fputc`; şeklinde bir ifade kullanıldığında;

- Bu komut sonucunda dosya yöneticisine bir çağrı olur. (Dosya yöneticisi işletim sisteminin I/O işlemlerini gerçekleştiren kısmıdır)
- İşletim sistemi byte'ın diske yazılıp yazılmadığından emin olur.
- Bir I/O işlemi için aşağıdaki işlemler gerekir.
 - Uygulama Programı (Application)
 - İşletim Sistemi (OS)
 - I/O İşlemcisi
 - Disk Kontrolcü (Disk Controller)

- **Uygulama Programı** : I/O işlemleri isteğinde bulunur.
- **İşletim Sistemi** : Tüm açık dosyaların bilgilerini tutar (dosya yöneticisi ile). Uygun sektörleri tampona (buffer) gönderir. Tempona byte yazar. I/O işlemcisi de tampondan verileri doğru, uygun disk bölgesine yazılması için komutlar verir.

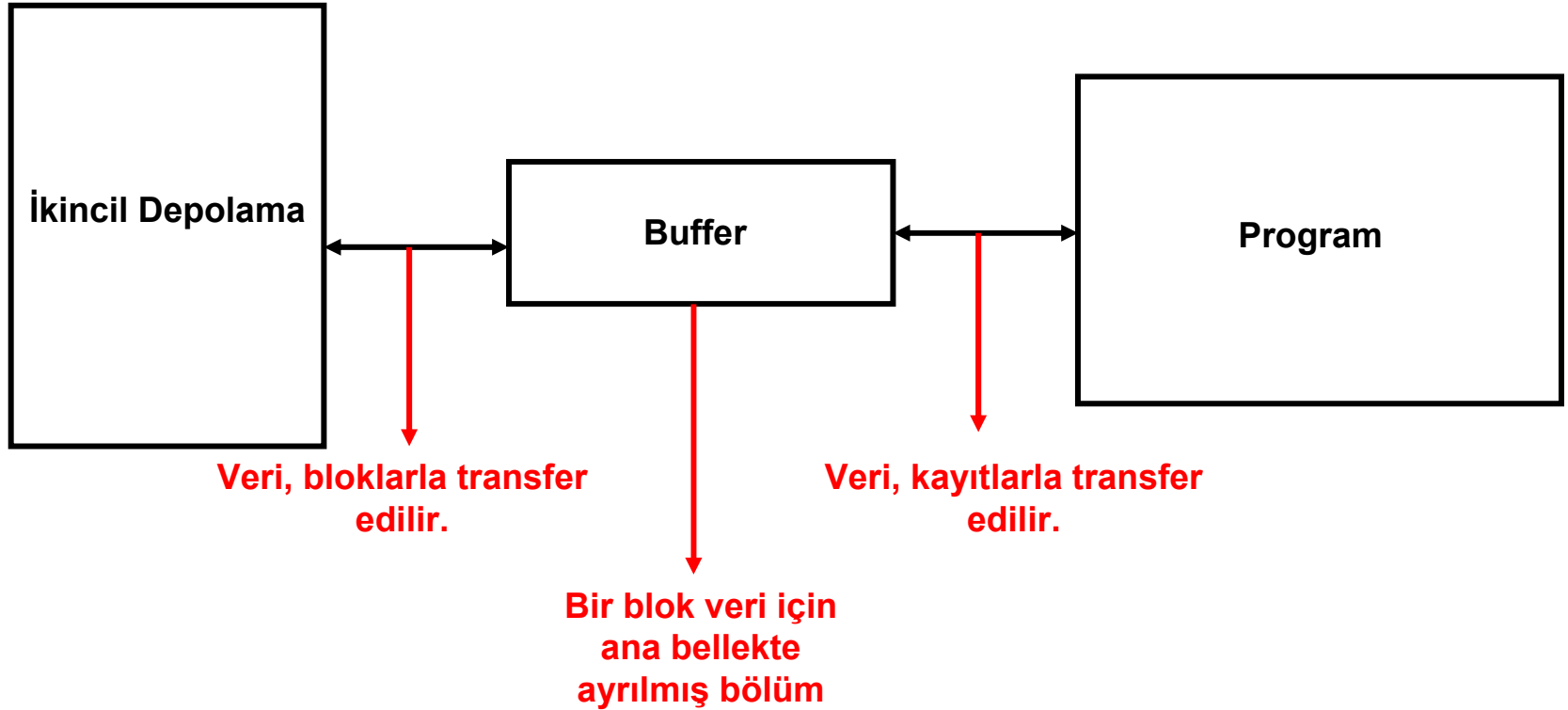
Not: Buffer bir cluster'ın 1-1 kopyasıdır.

- **I/O İşlemcisi** : Ayrı bir yongadır. CPU'dan bağımsız çalışır. Sürücüden uygun formatta veri almak ya da yazmak için sürücünün erişilebilir bir zamanı bulunur. Verileri disk kontrolcüsüne gönderir.
- **Disk Kontrolcüsü:** R/W kafasını yönetir.

Tampon Yönetimi (Buffer Management)

- Buffer'ın anlamı, bellekte olabildiğince büyük miktarda verilerle çalışmak ve bu sayede ikinci depolama aygıtlarına olan erişim sayısını düşürmektir.
- Sistem I/O buffer'dan bahsedeceğiz. Bu uygulama programların ötesinde işletim sistemi tarafından işletilen bir yapıdır.

- Uygulama programları kendi buffer'larını kullanabilirler. Bu bölgede büyük miktarda verileri diske daha sonra yazmak için biriktirir.



Sistem I/O Buffer

```
#include <stdio. h>
main()
{
FILE *file;
file=fopen("deneme","wt");
while (1)
    { fputc('A',file);
    }
fclose (file);
}
```

Böyle bir durumda
tek bir buffer
kullanılırsa ne olur?

Buffer Darboğazı

```
#include <stdio. h>
main()
{
FILE *file;
file=fopen("deneme","wt");
while (1)
    { fputc('A',file);
    }
fclose (file);
}
```

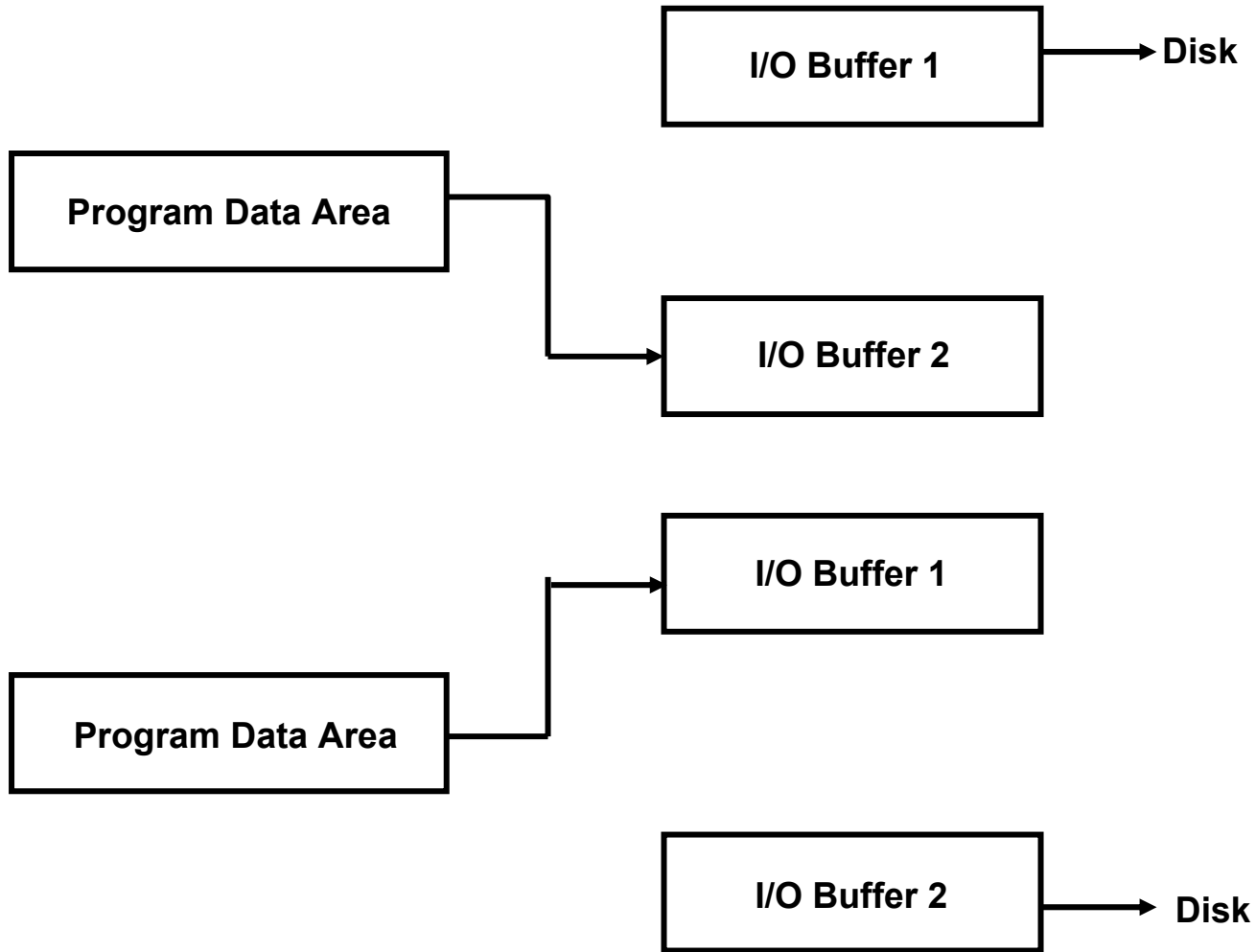
Buffer darboğazı oluşur.

Bir çok işletim sistemi bir giriş ve bir de çıkış buffer'ına sahiptir.

Buffer'lama Stratejileri

Hem işlem, hem de I/O işlemlerinin aynı zamanda yapılmasını sağlamak için iki buffer kullanılır. **Örn:** Varsayalım diske, sürekli veri yazan bir uygulama programı olsun.

- İşlemci, buffer'ı I/O işlemi gerçekleşirken doldurmak isteyebilir.
- Eğer 2 buffer kullanılırsa I/O-CPU örtüşmesine izin verilir. Yani CPU, bir buffer diske transfer edilirken diğerini doldurabilir.
- Her iki görev bittiğinde bufferların rolleri değiştirilebilir.
- Gerçekte bu yönetim işletim sistemi tarafından yapılır.



2'li Buffer Kullanımı

Diğer Stratejiler

- **Çoklu Bufferlama:** Birden fazla sayıda buffer kullanılabilir.
- **Buffer Havuzu:** Bir buffer havuzu bulunur.
 - Bir sektör için bir istek geldiğinde bu sektörün bu bufferlar içinde olup olmadığına bakılır.
 - Yoksa sektör okunarak havuzdaki boş bir buffer'a alınır. Boşta buffer yoksa bufferlardan biri boşaltılır.