

User



Kernel



Dispozitive de Intrare/IeSire

```
int fd = open / socket / pipe / ...;  
read(fd, buf, 10);  
write(fd2, buf, 10);  
close(fd);
```

Dispozitive de I/O read/write discrete		
0	1	2
3	4	5
6	7	8
9	10	11
12	13	14
15	16	17
18	19	20
21	22	23
24	25	26
27	28	29
30	31	32
33	34	35
36	37	38
39	40	41
42	43	44
45	46	47
48	49	50
51	52	53
54	55	56
57	58	59
60	61	62
63	64	65
66	67	68
69	70	71
72	73	74
75	76	77
78	79	80
81	82	83
84	85	86
87	88	89
90	91	92
93	94	95
96	97	98
99	100	101
102	103	104
105	106	107
108	109	110
111	112	113
114	115	116
117	118	119
120	121	122
123	124	125
126	127	128
129	130	131
132	133	134
135	136	137
138	139	140
141	142	143
144	145	146
147	148	149
150	151	152
153	154	155
156	157	158
159	160	161
162	163	164
165	166	167
168	169	170
171	172	173
174	175	176
177	178	179
180	181	182
183	184	185
186	187	188
189	190	191
192	193	194
195	196	197
198	199	200
201	202	203
204	205	206
207	208	209
210	211	212
213	214	215
216	217	218
219	220	221
222	223	224
225	226	227
228	229	230
231	232	233
234	235	236
237	238	239
240	241	242
243	244	245
246	247	248
249	250	251
252	253	254
255	256	257
258	259	260
261	262	263
264	265	266
267	268	269
270	271	272
273	274	275
276	277	278
279	280	281
282	283	284
285	286	287
288	289	290
291	292	293
294	295	296
297	298	299
299	300	301
302	303	304
305	306	307
308	309	310
311	312	313
314	315	316
317	318	319
320	321	322
323	324	325
326	327	328
329	330	331
332	333	334
335	336	337
338	339	340
341	342	343
344	345	346
347	348	349
350	351	352
353	354	355
356	357	358
359	360	361
362	363	364
365	366	367
368	369	370
371	372	373
374	375	376
377	378	379
380	381	382
383	384	385
386	387	388
389	390	391
392	393	394
395	396	397
398	399	400
401	402	403
404	405	406
407	408	409
410	411	412
413	414	415
416	417	418
419	420	421
422	423	424
425	426	427
428	429	430
431	432	433
434	435	436
437	438	439
440	441	442
443	444	445
446	447	448
449	450	451
452	453	454
455	456	457
458	459	460
461	462	463
464	465	466
467	468	469
470	471	472
473	474	475
476	477	478
479	480	481
482	483	484
485	486	487
488	489	490
491	492	493
494	495	496
497	498	499
499	500	501
502	503	504
505	506	507
508	509	510
511	512	513
514	515	516
517	518	519
520	521	522
523	524	525
526	527	528
529	530	531
532	533	534
535	536	537
538	539	540
541	542	543
544	545	546
547	548	549
550	551	552
553	554	555
556	557	558
559	560	561
562	563	564
565	566	567
568	569	570
571	572	573
574	575	576
577	578	579
580	581	582
583	584	585
586	587	588
589	590	591
592	593	594
595	596	597
598	599	600
601	602	603
604	605	606
607	608	609
610	611	612
613	614	615
616	617	618
619	620	621
622	623	624
625	626	627
628	629	630
631	632	633
634	635	636
637	638	639
640	641	642
643	644	645
646	647	648
649	650	651
652	653	654
655	656	657
658	659	660
661	662	663
664	665	666
667	668	669
670	671	672
673	674	675
676	677	678
679	680	681
682	683	684
685	686	687
688	689	690
691	692	693
694	695	696
697	698	699
699	700	701
702	703	704
705	706	707
708	709	710
711	712	713
714	715	716
717	718	719
720	721	722
723	724	725
726	727	728
729	730	731
732	733	734
735	736	737
738	739	740
741	742	743
744	745	746
747	748	749
750	751	752
753	754	755
756	757	758
759	760	761
762	763	764
765	766	767
768	769	770
771	772	773
774	775	776
777	778	779
780	781	782
783	784	785
786	787	788
789	790	791
792	793	794
795	796	797
798	799	800
801	802	803
804	805	806
807	808	809
810	811	812
813	814	815
816	817	818
819	820	821
822	823	824
825	826	827
828	829	830
831	832	833
834	835	836
837	838	839
840	841	842
843	844	845
846	847	848
849	850	851
852	853	854
855	856	857
858	859	860
861	862	863
864	865	866
867	868	869
870	871	872
873	874	875
876	877	878
879	880	881
882	883	884
885	886	887
888	889	890
891	892	893
894	895	896
897	898	899
899	900	901
902	903	904
905	906	907
908	909	910
911	912	913
914	915	916
917	918	919
920	921	922
923	924	925
926	927	928
929	930	931
932	933	934
935	936	937
938	939	940
941	942	943
944	945	946
947	948	949
950	951	952
953	954	955
956	957	958
959	960	961
962	963	964
965	966	967
968	969	970
971	972	973
974	975	976
977	978	979
980	981	982
983	984	985
986	987	988
989	990	991
992	993	994
995	996	997
998	999	999

Dispozitivele de I/O Sunt foarte diverse

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read-write	CD-ROM graphics controller disk



Ce vor utilizatori?

Usurinta in utilizare:

- cod care ruleaza pe oricate dispozitive de intrare-iesire
- sa fie usor sa numeasca dispozitivele
- tratarea erorilor sa se intample automat

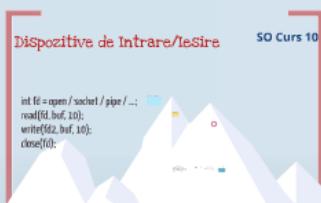
Performanta

Operatii dorite:

- sincrone (majoritatea)
- asincrone (unele aplicatii)

Cum implementam?

User



User level Software

Device Independent Software

Opereaza independent de dispozitive

- Probleme specifice hardware
- Rezolvare problema hardware
- Rezolvare problema software
- Resurse disponibile

Opereaza independent de dispozitive

- Deschide fisier la nivel de utilizator
- Scrie in fisier
- Deschide fisier
- Resurse disponibile

Device Drivers

Polling
Intreruperi
Direct-memory access

Cum folosim hardware?

Intreruperi

Hardware

Data categorii de dispozitive care au si box:
Fisier de categorie:

- dispozitive de comunicare
- unitati de memorare
- unitati de lucru cu harti
- unitati de lucru magnetice

Domeniu de cumparaturi de IT din Romania:
Categoriile de gestionarea dispozitivelor

Kernel



Prezi

Hardware

Doua categorii de dispozitive: caracter si bloc

Fiecare dispozitiv are:

- un controller
- unul sau mai multe porturi
- conectat la o magistrala

Procesorul da comenzi de I/O controllerului

Controllerul gestioneaza dispozitivul



Comunicația cu dispozitivele de I/O

- regiile de stare control, informe, ieșire
- (optinental) într-o parte dintr-o date

• Date referitoare un dispozitiv de I/O

Port mapped I/O Memory mapped I/O

Interfață

Doua categorii de dispozitive: caracter si bloc

Fiecare dispozitiv are:

- un controller
- unul sau mai multe porturi
- conectat la o magistrala

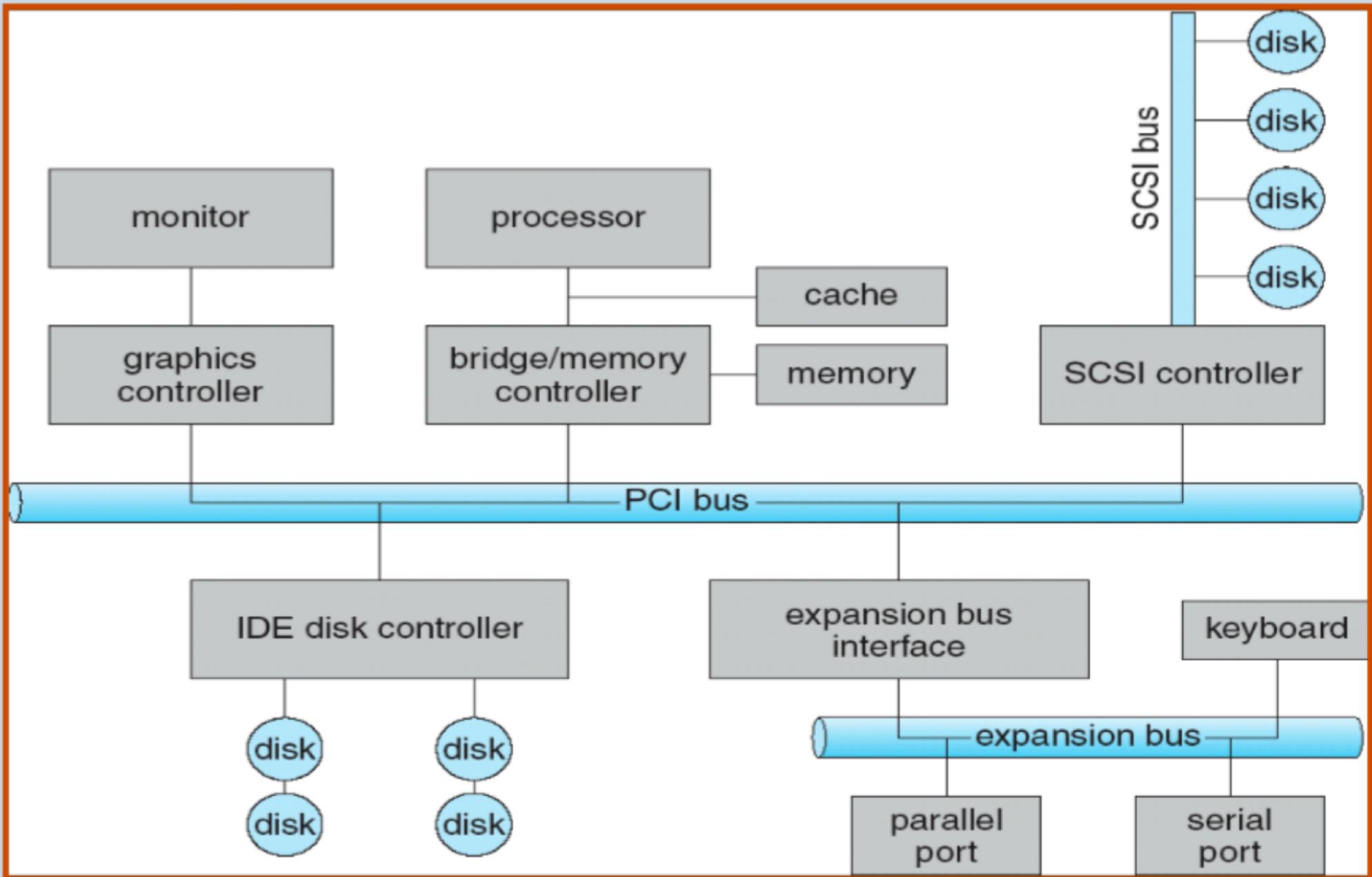
Procesorul da comenzi de I/O controllerului

Controllerul gestioneaza dispozitivul

Comunica
Fiecare controller:
- registre
- (optional)
Cum ac

Port-mapped





Comunicatia cu dispozitivele de I/E

Fiecare controller are:

- registre de stare, control, intrare, iesire
- (optional) buffer pentru date

Cum adresam un dispozitiv de I/O?

Port-mapped I/O

Spatiu de adresa separat pentru dispozitivele de I/O

O adresa de I/O se numeste **port**

Un port adreseaza un regisztrul controllerului

instructiuni specializate:

INP, OUT

REG, 0x2FA

Memory mapped I/O

Registrele I/O sunt mapate in spatiul de memorie

Avantaje:

- nu necesita instructiuni specializate
- protectia de la memoria virtuala
- instructiunile pot referi memorie sau registre

Dificultati:



Port-mapped I/O

Spatiu de adresa separat pentru dispozitivele de I/O

O adresa de I/O se numeste **port**

Un port adreseaza un registru al controllerului

Instructiuni specializate:

- IN REG, 0x2F8
- OUT REG, 0x2FA

Memory mapped I/O

Registrele I/O sunt mapate în spațiul de memorie

Avantaje:

- nu necesita instrucțiuni specializate
- protecția de la memoria virtuală
- instrucțiunile pot referi memorie sau registre

Dezavantaje:

- trebuie inhibat cache-ul la nivel de pagina
- bridge-uri între magistrale -> mai lent decât port-mapped I/O

User



Dispozitive de Intrare/Iesire

SO Curs 10

```
int fd = open / socket / pipe / ...  
read(fd, buf, 10);  
write(fd, buf, 10);  
close(fd);
```

USER level Software



Device Independent Software

Operatii independent de dispozitiv
• Redirecție la device specific
• Buffering
• Comprimarea și decompresia date
• Deschidere/închidere
• Încărcare/șters

Operatii independent de dispozitiv (II)
• Salvează date în cache
• Specificațiile de la nivel de utilizator
• Utilizare directă

Threading
• Multithreading
• Threadafe

Device Drivers



Polling
Intreruperi

Direct-memory access

Cum folosim hardware?

Intreruperi

Hardware

• Dispozitive de caracteristică sticla
• Noduri de rețea
• I/O controller
• Unități de memorie portante
• Conectare la rețea
• Procesorul de comandă din unitatea de lucru
• Controlerul de gestionare a dispozitivelor

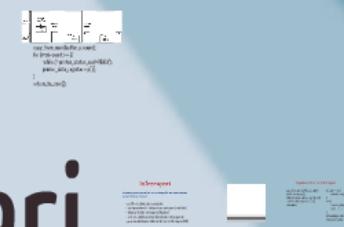
Kernel

vers

Polling

Intreruperi

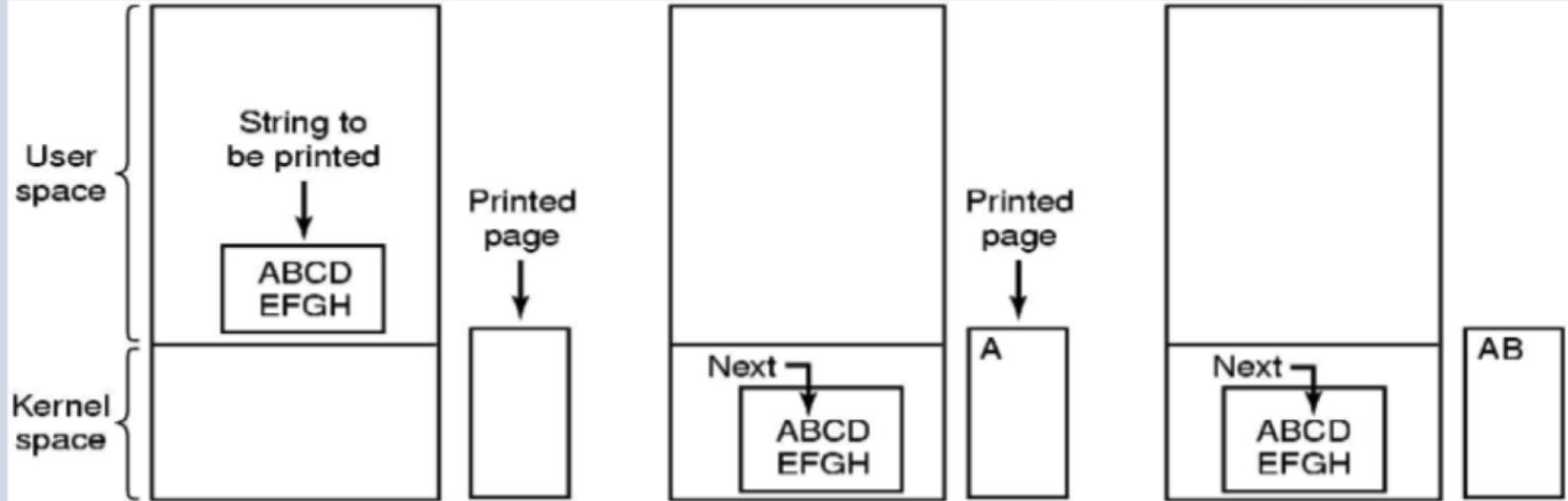
Direct-memory access



Direct Memory Access

DMA: dispozitiv separat specializat pentru copieri din memorie in memorie
CPU comanda DMA sa efectueze transferuri date ca (sursa,destinatie,octeti)
CPU nu poate accesa magistrala daca este folosita de DMA

Cum folosim hardware?



```

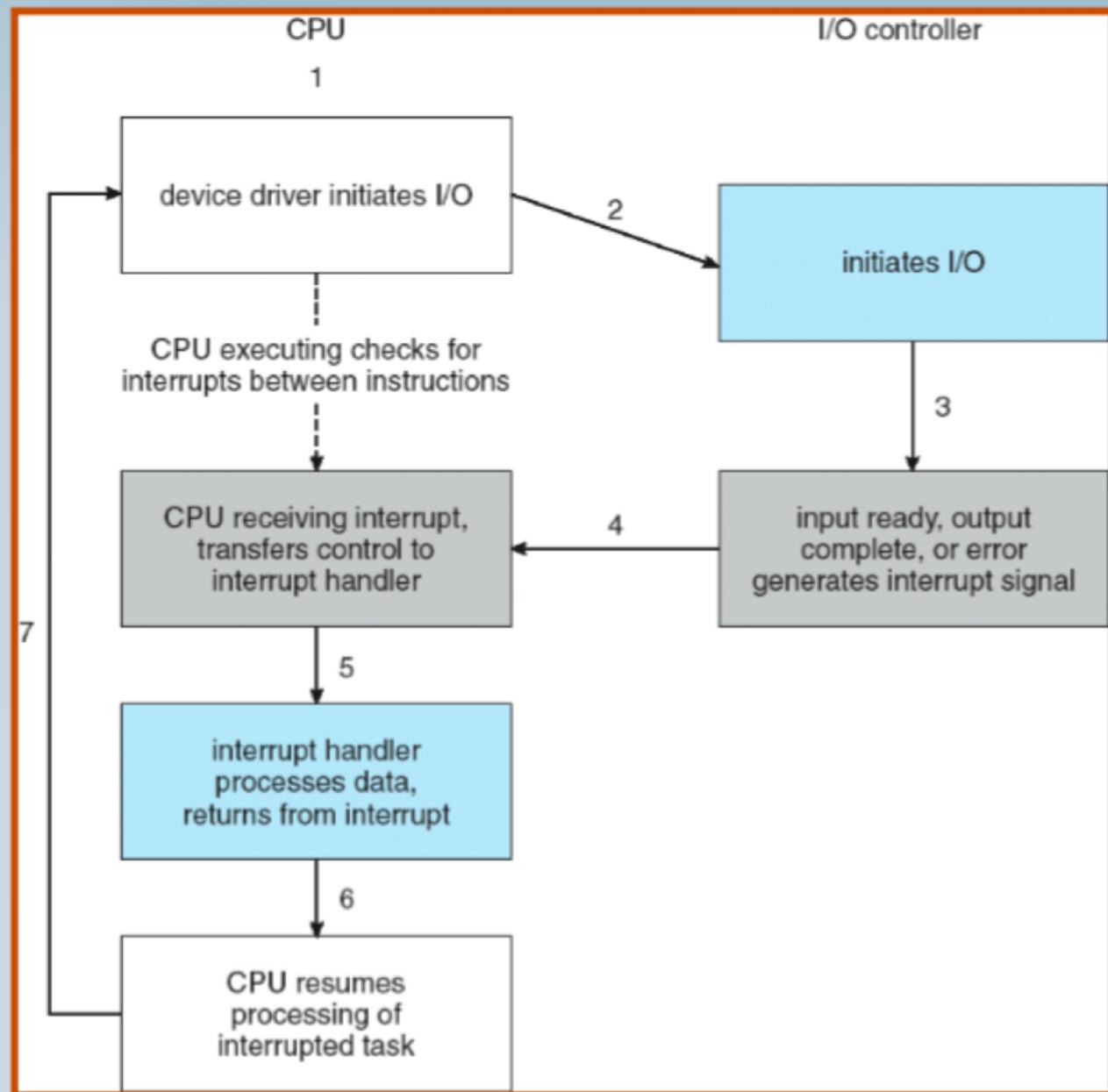
copy_from_user(buffer, p, count);
for (i=0;i<count;i++){
    while (*printer_status_req!=READY);
    printer_data_register = p[i];
}
return_to_user();

```

Intreruperi

Anunta procesorul ca s-a intamplat un eveniment care trebuie tratat

- pot fi mascabile sau nemascabile
- sunt generate de software sau hardware (controller)
- folosesc linii de interrupere (IRQ line)
- exista o tabela cu rutine de tratare a interruperilor
- procesorul ruleaza rutina de tratare a interruperii (ISR)



Implementare cu intreruperi

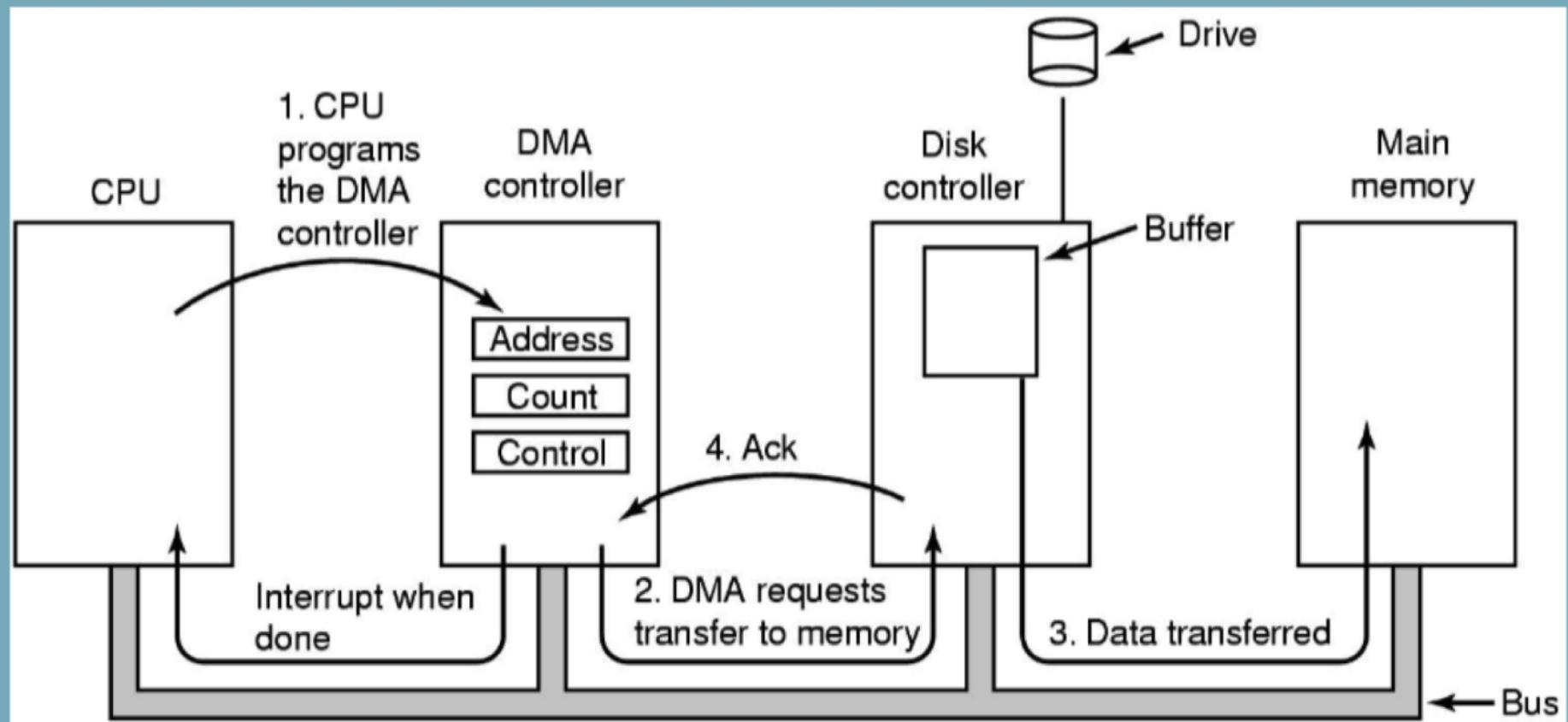
```
copy_from_user(buffer, p , count);
enable_interrupts();
while (*printer_status_reg!=READY);
*printer_data_register = p[0];
scheduler();
if (count==0) {
    unblock_user();
}
else {
    *printer_data_register = p[i];
    count --; i++;
}
acknowledge_interrupt();
return_from_interrupt;
```

Direct Memory Access

DMA: dispozitiv separat specializat pentru copieri din memorie in memorie

CPU comanda DMA sa efectueze transferuri date ca (sursa,destinatie,octeti)

CPU nu poate accesa magistrala daca este folosita de DMA



Implementare cu DMA

```
copy_from_user(buffer,p,count);    acknowledge_interrupt();  
set_up_DMA_controller();          unblock_user();  
scheduler();                      return_from_interrupt();
```

Device Independent Software

Operatii independente de dispozitiv

Planificare pentru eficienta si echitate:

- sortare si comasare de cereri de I/O

Buffering

- Compensarea vitezei de transfer dintre dispozitive
- Reasamblarea datelor
- Semantica de copiere pentru utilizatori

Operatii independente de dispozitiv (2)

Caching: memoria tine copii ale datelor

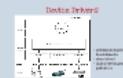
Spooling: se mentine un buffer cu datele transmise unui dispozitiv

Rezervarea dispozitivului



Fluxul unei operatii de I/O

Device Drivers



Polling
Intreruperi

Direct-memory access

Cum folosim hardware?



Direct Memory Access

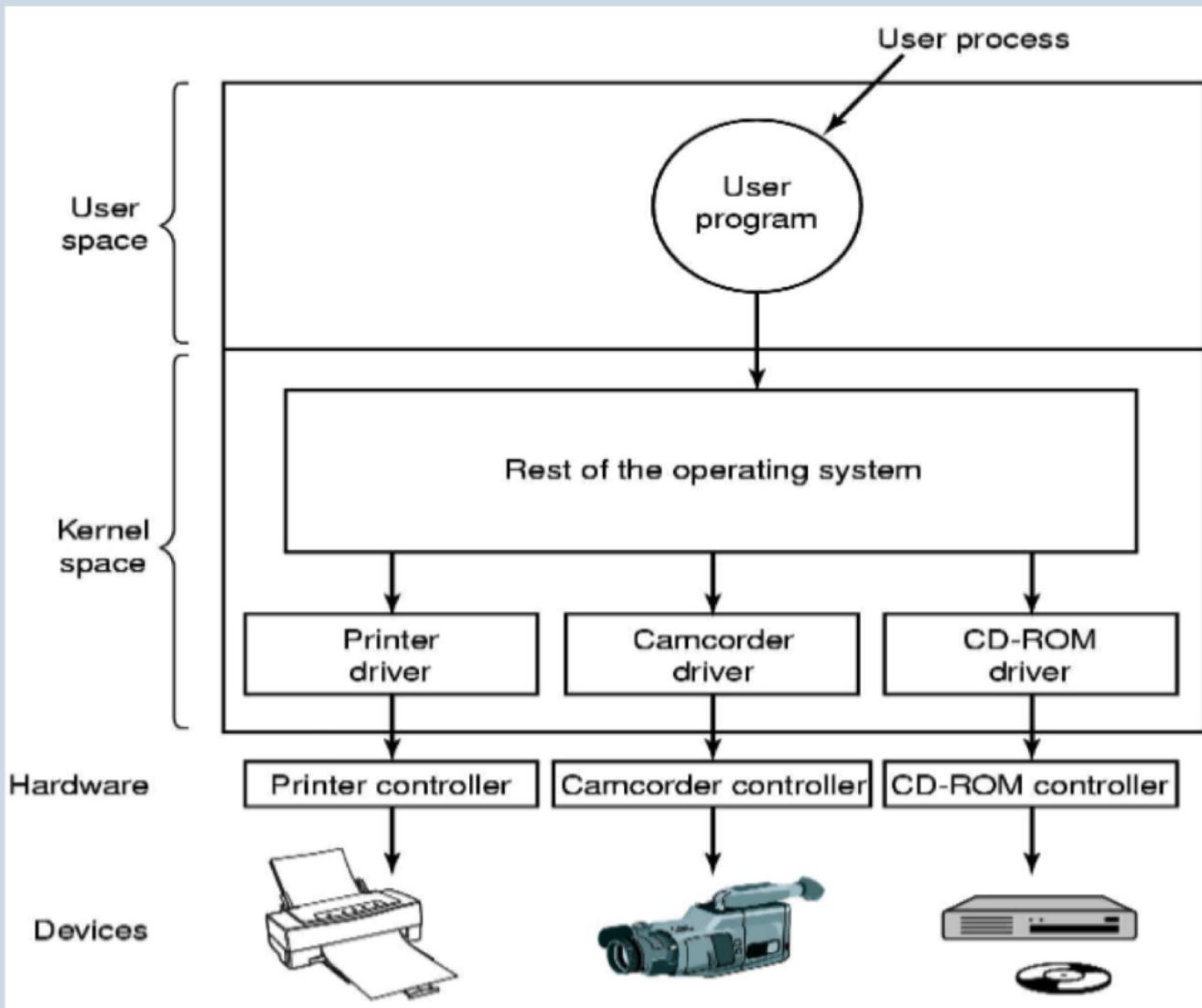
DMA desparte segmentul de lucru pentru procesor (CPU) de segmentul de lucru pentru memorie (memorie RAM). În urma acestei diferențe, datele sunt transmise din RAM la periferice și nu trebuie să fie procesate de CPU.

Depărtarea cu DMA

Împărțirea perifericelor între CPU și DMA. Aceasta permite să se proceseze datele în paralel.

Intreruperi

Device Drivers



- controleaza un dispozitiv sau clasa de dispozitive
- ruleaza in kernel
- SO ofera interfata comună pentru drivere

User level Software

Device Independent Software

Operatii independente de dispozitiv

Planificare pentru eficiență și echitate:

- sortare și comasare de cereri de I/O

Buffering

- Compensarea vitezelor de transfer dintre dispozitive
- Reasamblarea datelor
- Semantica de copiere pentru utilizatori

Operatii independente de dispozitiv (2)

Caching: memoria tine copii ale datelor

Spooling: se menține un buffer cu datele transmise unui dispozitiv

Reservarea dispozitivului

Fluxul unei operatii de I/O



Device Drivers

Polling

Intreruperi

Direct-memory access



Small Memory Access
DMA devices can't specify part of memory
DMA controller is responsible for moving data
DMA controller has to be programmed by software
DMA controller manages the data with help of DMA



Independent DMA
DMA controller is part of北桥 (Northbridge)
DMA controller manages memory access

Cum folosim hardware?

Intreruperi



Prezi

Operatii independente de dispozitiv

Planificare pentru eficienta si echitate:

- sortare si comasare de cereri de I/O

Buffering

- Compensarea vitezei de transfer dintre dispozitive
- Reasamblarea datelor
- Semantica de copiere pentru utilizatori

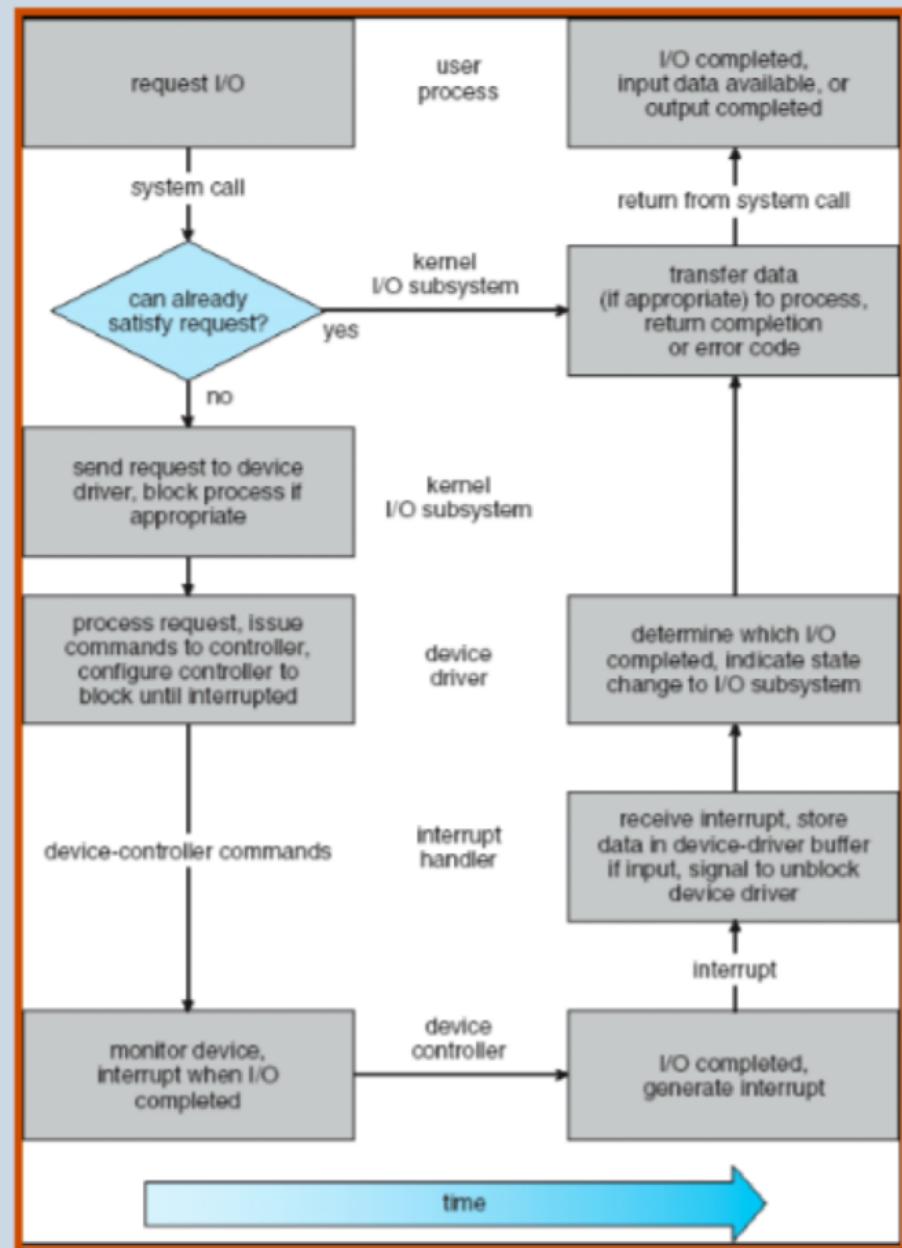
Operatii independente de dispozitiv (2)

Caching: memoria tine copii ale datelor

Spooling: se mentine un buffer cu datele transmise unui dispozitiv

Rezervarea dispozitivului

Fluxul unei operatii de I/O



User level Software

Device Independent Software

Operatii independente de dispozitiv

- Planificare pentru eficiență a solicitărilor
- sortare și organizare de cureau de I/O
- Buffering:
 - Compartimente fizice de transfer dintre dispozitive
 - Redămândarea datele
 - Semantica de copiere pentru utilizatori

Operatii independente de dispozitiv (2)

- Coduri: memoria în care se copiază datele
- Spooling: se menține un buffer cu datele transmise unui dispozitiv
- Reservarea dispozitivelor

Fluxul unei operatii de I/O



Device Drivers

Polling

Intreruperi

Direct-memory access



Cum folosim hardware?

Intreruperi

Hardware

Două categorii de dispozitive: caracter și bloc

Fiecare dispozitiv are:

- un controller
- unul sau mai multe porturi
- conectată la o magistrală

Procesorul da comenzi de I/O controllerului

Controllerul gestionează dispozitivul



Prezi

API Utilizator

Char Devices

e.g. mouse, tastatura

Acces sequential

Operatii: put, get

Transfer la nivel de caracter

Viteza redusa

Block Devices

e.g. discuri, CDROM

Acces aleator

Operatii: read, write, seek

Transfer la nivel de bloc

Viteza ridicata

Network

e.g. 802.11, 802.5

Separare protocol de interfata

Sockets

Operatii: connect, send, recv, close

Viteza ridicata

Initializare

**Dispozitive block/character
open, close**

Dispozitive retea

socket, connect, shutdown

Control

Controleaza dispozitivul de I/O
Depinde de dispozitiv!

- ioctl / DeviceIOControl
- setsockopt, getsockopt

Tipuri de operatii I/O

Blocante

Procesul este suspendat pana la incheierea operatiei

Simplu de folosit

Ne-blocante

Operatia se intoarce imediat

Procesul primeste datele disponibile

Asincrone

Procesul ruleaza in paralel cu operatia

Este notificat atunci cand operatia este finalizata

Event-driven programming

Tipuri de operatii I/O

	blocante	ne-blocante
sincrone	read/write ReadFile, WriteFile	read/write O_NONBLOCK
asincrone	select	AIO aio_*, overlapped I/O

ASynchronous I/O

Unix

aio_read

aio_write

aio_suspend

Window

ReadFile

WriteFile

OVERLAPPED I/O

User



Dispozitive de Intrare/Iesire

SO Curs 10

```
int fd = open / socket / pipe / ...;  
read(fd, buf, 10);  
write(fd, buf, 10);  
close(fd);
```

USeR level Software

Device Independent Software

Operatiuni independente de dispozitive

- Părăsirea portului de utilizator
- Crearea și închiderea unei clări
- Închidere
- Crearea și închiderea unei fișiere
- Citire și scriere
- Setarea unor proprietăți

Operatiuni independente de dispozitive I/O

- Citirea și scrierea unei valori de la/într-un dispozitiv
- Sprijinirea unei valori cu datele transmise prin I/O
- Afisarea unor date

Plasarea operațiunii de I/O

Afisarea unor date

Device Drivers

Polling
Intreruperi
Direct-memory access

Cum folosim hardware?

Intreruperi

Hardware

Două categorii de dispozitive: caracter si bloc

- un controller
- unul sau mai multe porturi
- conectat la o magistrală

Procesorul documentează I/O controller-ului

Controllerul gestionează dispozitivul

Kernel

