

ЛЕКЦ 1

Үйлдлийн системийн онол

Хичээл заах багш: МТ –н салбарын
доктор, профессор О. Бат-Энх

ҮЙЛДЛИЙН СИСТЕМИЙН ОНОЛ

Бүлэг 1 Танилцуулга

Доктор, профессор О. Бат-Энх

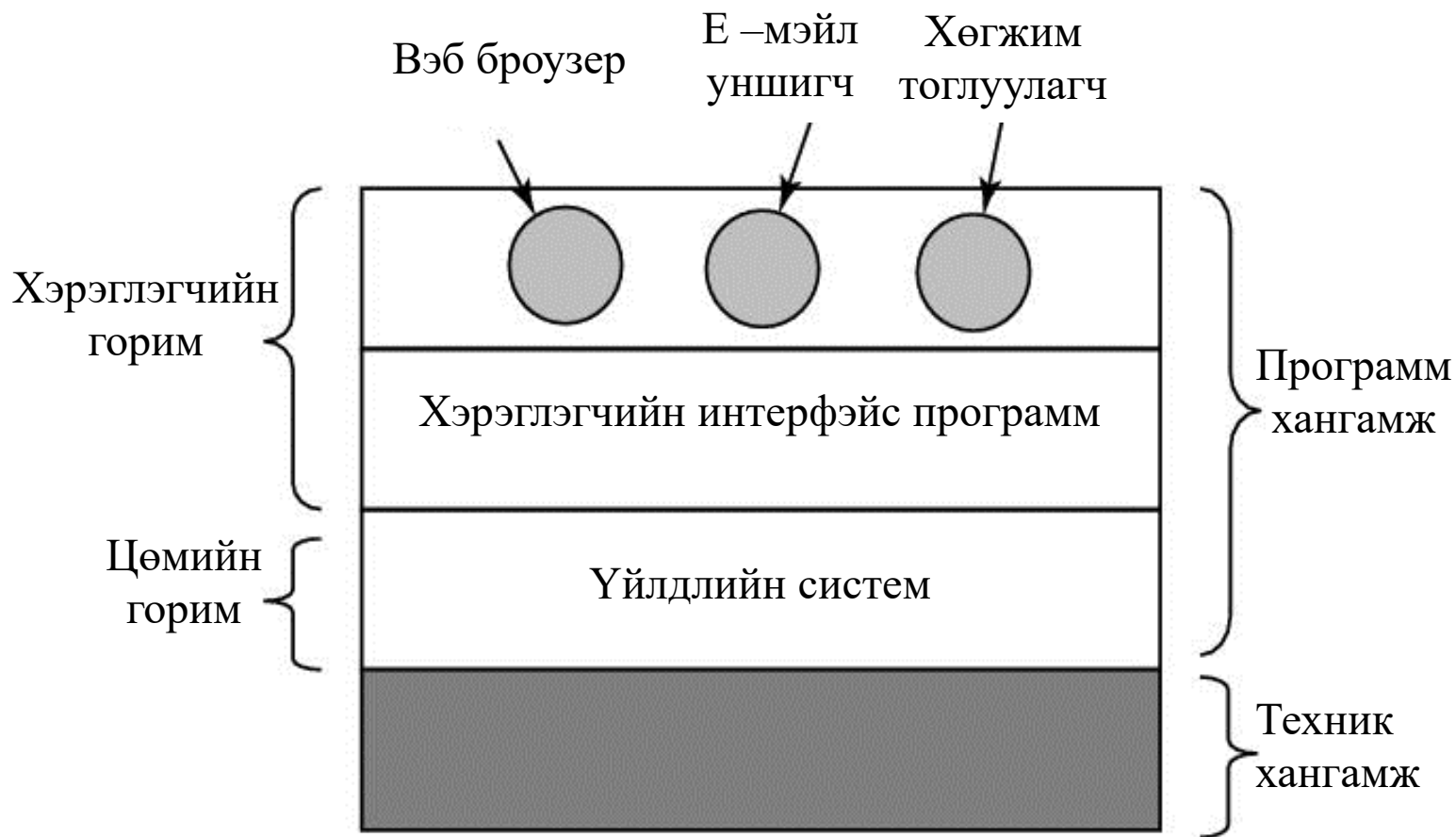
Үйлдлийн систем (1)

Орчин үеийн компьютер:

- Нэг ба түүнээс олон процессор
- Үндсэн санах ой
- Дискнүүд
- Принтерүүд
- Төрөл бүрийн оролт /гаралтын төхөөрөмжүүд

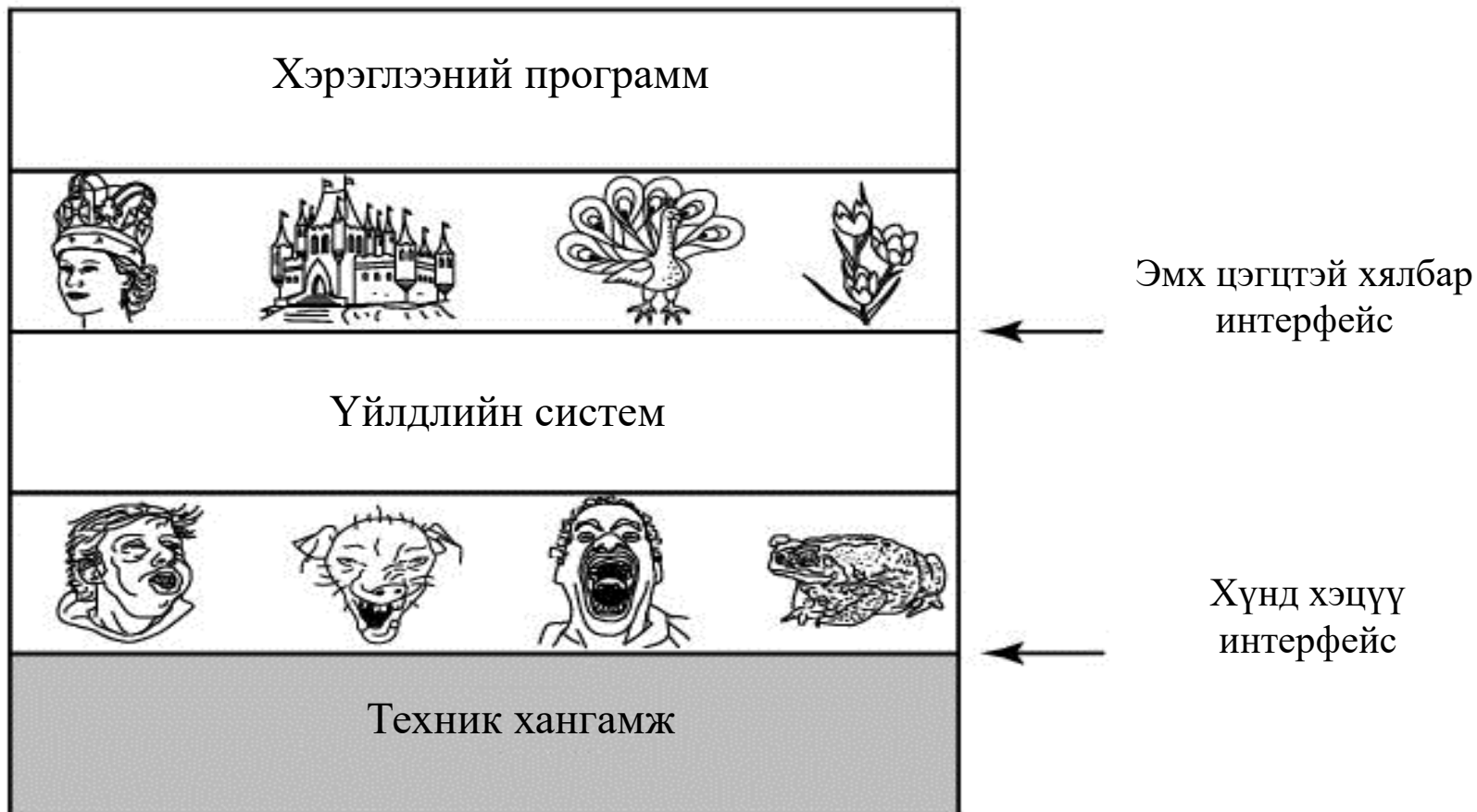
Эдгээр бүрэлдэхүүн хэсгүүдийг удирдах програм хангамж –**үйлдлийн систем**

Үйлдлийн систем (2)



Зураг 1-1. Үйлдлийн системийн байрлал

Өргөтгөсөн машин (ҮС)



Зураг 1-2. Үйлдлийн системүүд хэцүү техник хангамжаас эмх цэгцтэй хийсвэр рүү байдал руу шилжиж байна.

Нөөцийн удирдлага

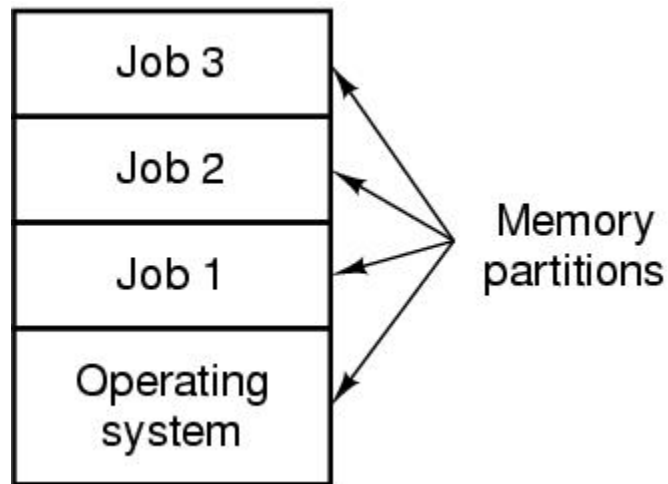
- Нэг зэрэг олон программ ажиллахыг зөвшөөрөх
- Санах ой, О/Г –ын төхөөрөмжүүд ба бусад нөөцүүдийг удирдах ба хамгаалах
- Нөөцийг хамтран эзэмших 2 арга байна.
 - Хугацаа
 - Зай

Үйлдлүүдийн системүүдийн тухай

Хөгжил:

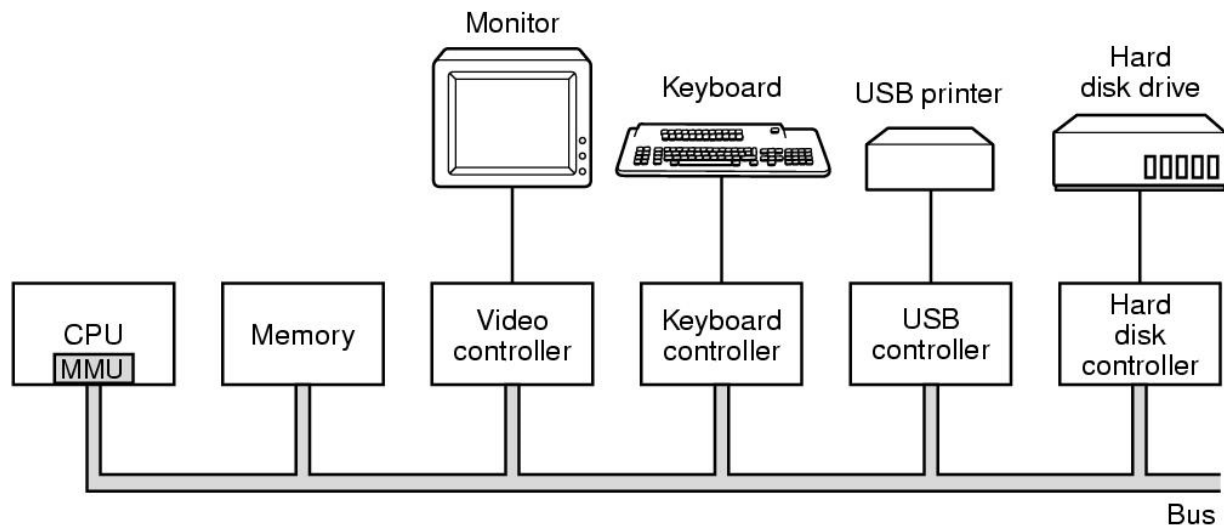
- (1945–55) Vacuum Tubes
- (1955–65) Transistors and Batch Systems
- (1965–1980) ICs and Multiprogramming
- (1980–Одоог хүртэл) Personal Computers

IC ба олон программчлал



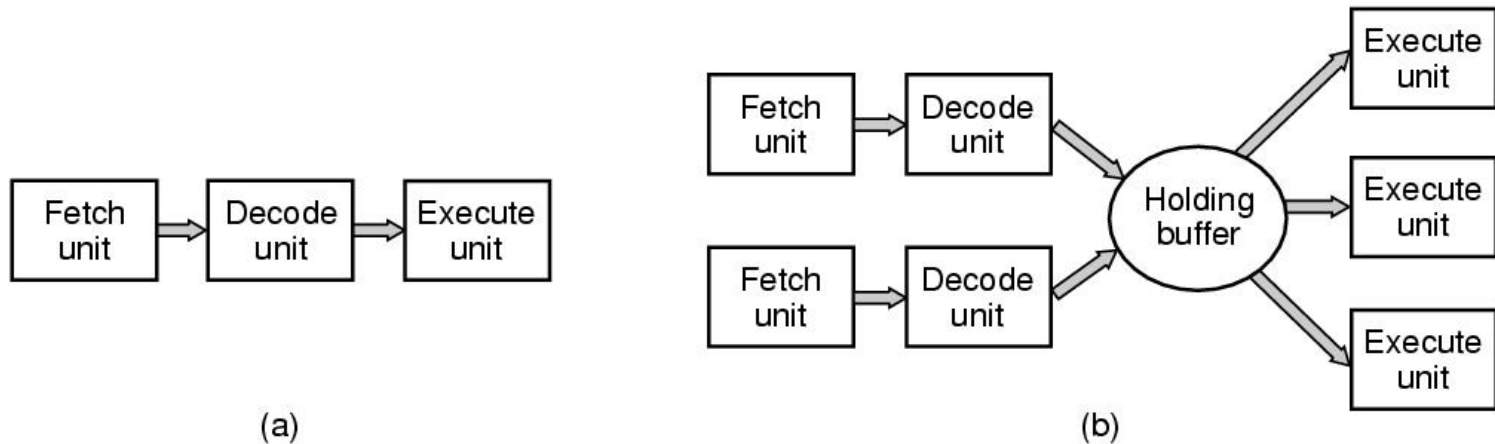
Зураг 1-5. Санах ойд байрлах 3 ажилтай олон программын систем.

Компьютерын техник хангамжийн тухай



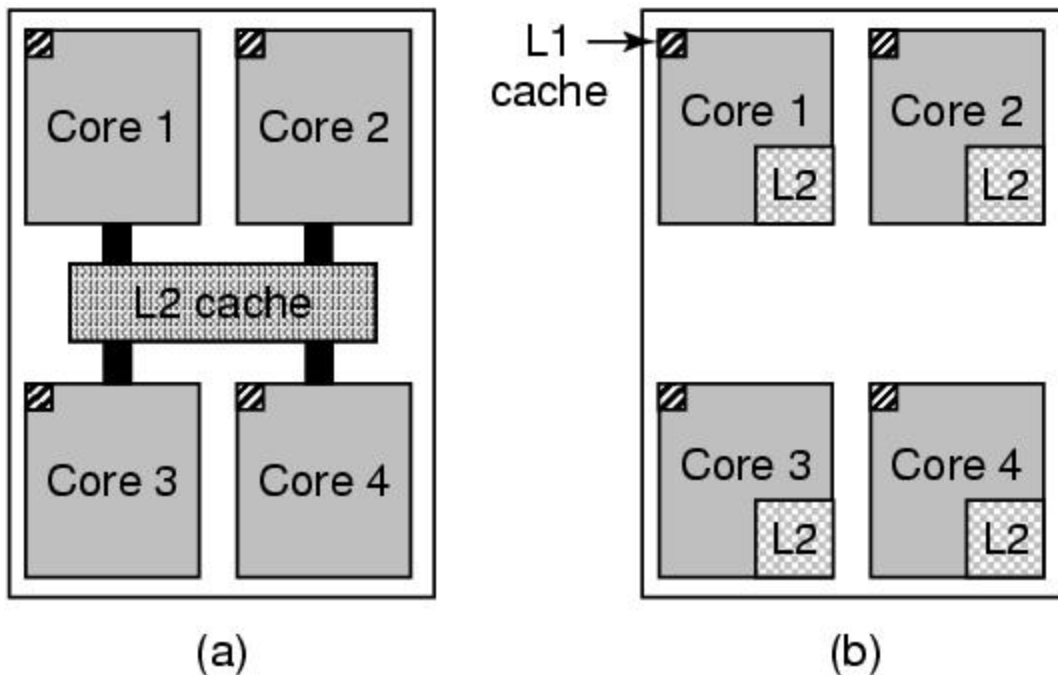
Зураг 1-6. Энгийн персонал компьютерын бүрэлдэхүүн хэсгүүд.

CPU Pipelining



Зураг 1-7. (a) 3 үе шаттай pipeline. (b) A superscalar CPU.

Олон салбар процесс ба олон цөмийн ЧИПҮҮД



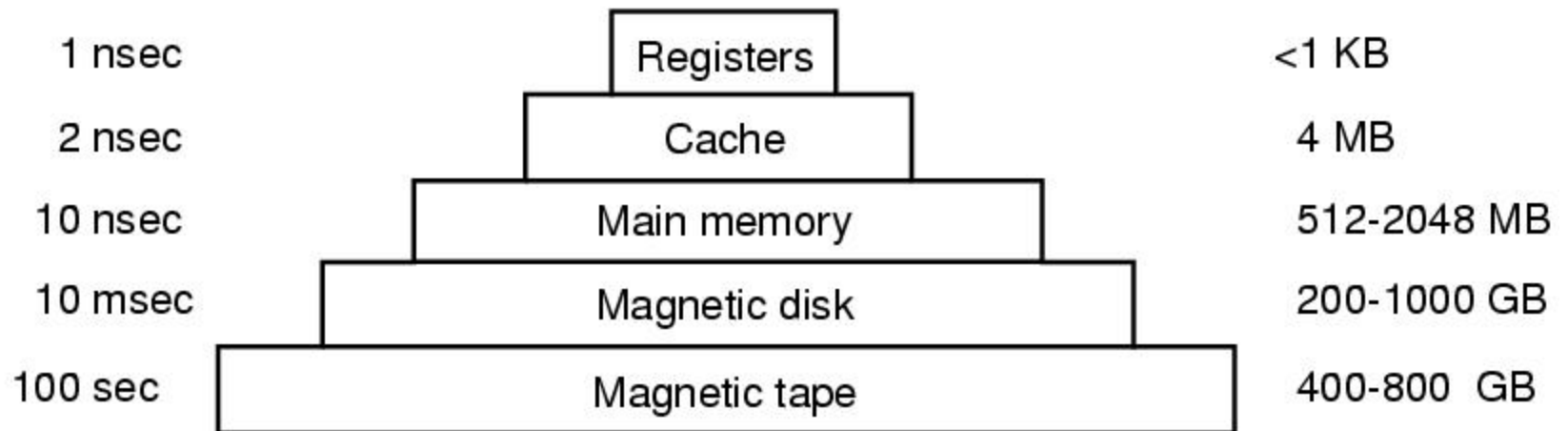
Зураг 1-8. (a) 4 цөмтэй чип (хамтран эзэмшиж байгаа L2 кэштэй).

(b) 4 цөмтэй чип (L2 кэшүүд биеэ даасан тусдаа)

Санах ой (1)

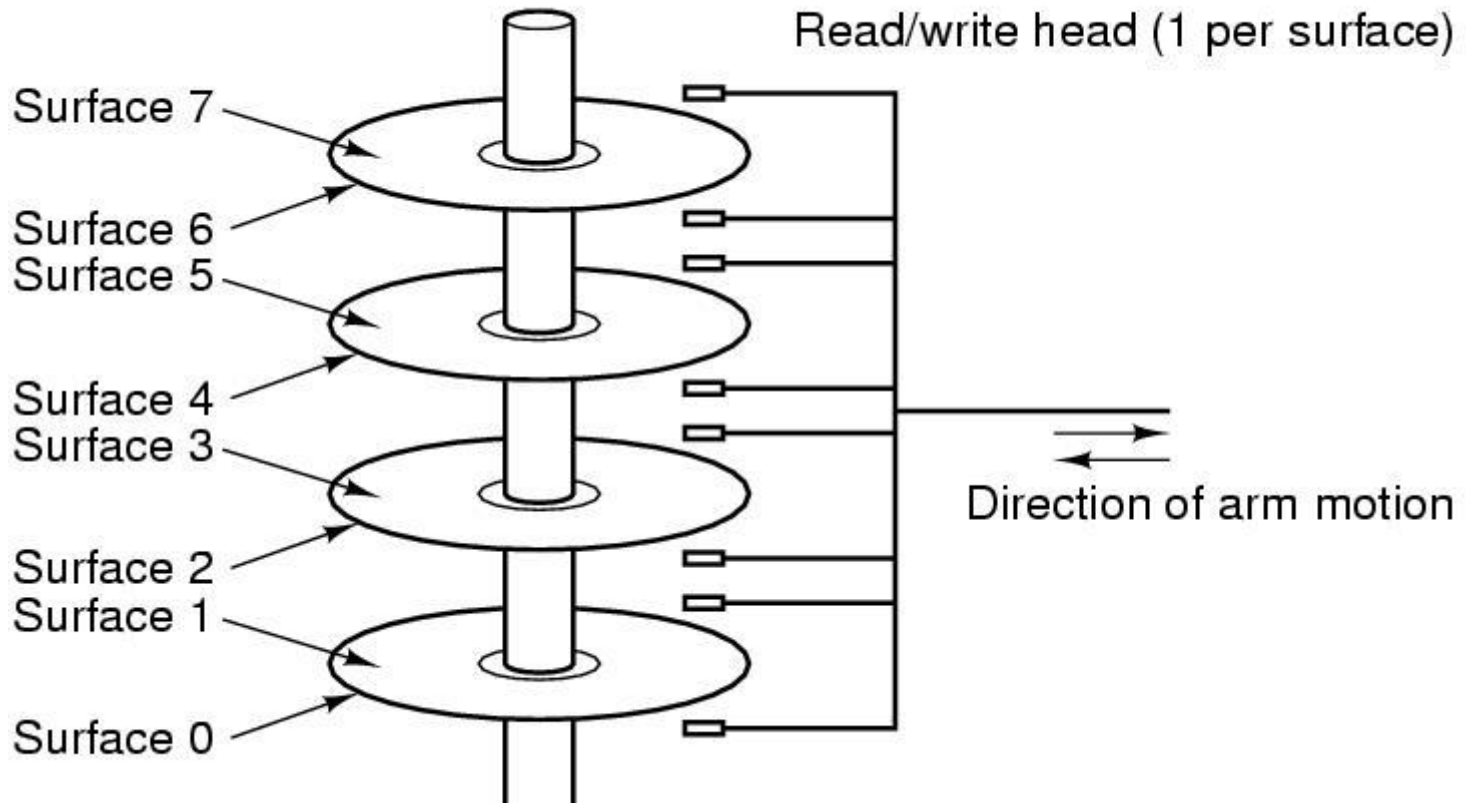
Typical access time

Typical capacity



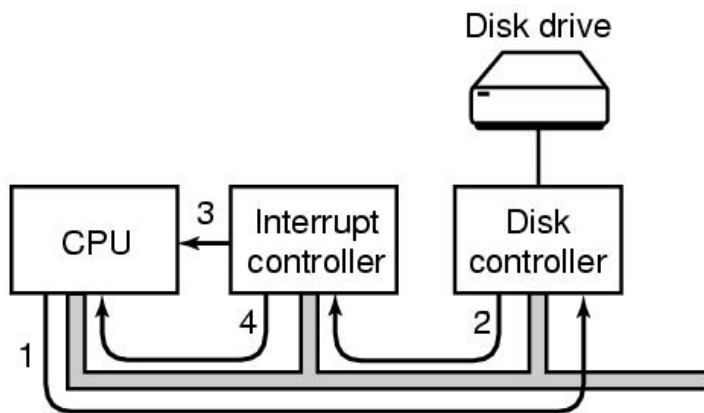
Зураг 1-9. Санах ойн шатлалын төрөл.

ДИСКНҮҮД

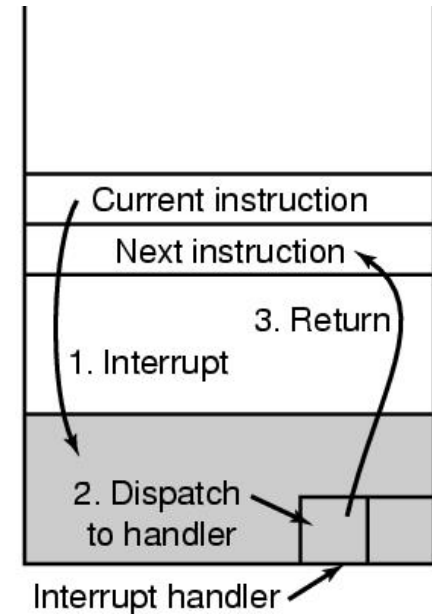


Зураг 1-10. Диск драйверын бүтэц.

О/Г төхөөрөмжүүд



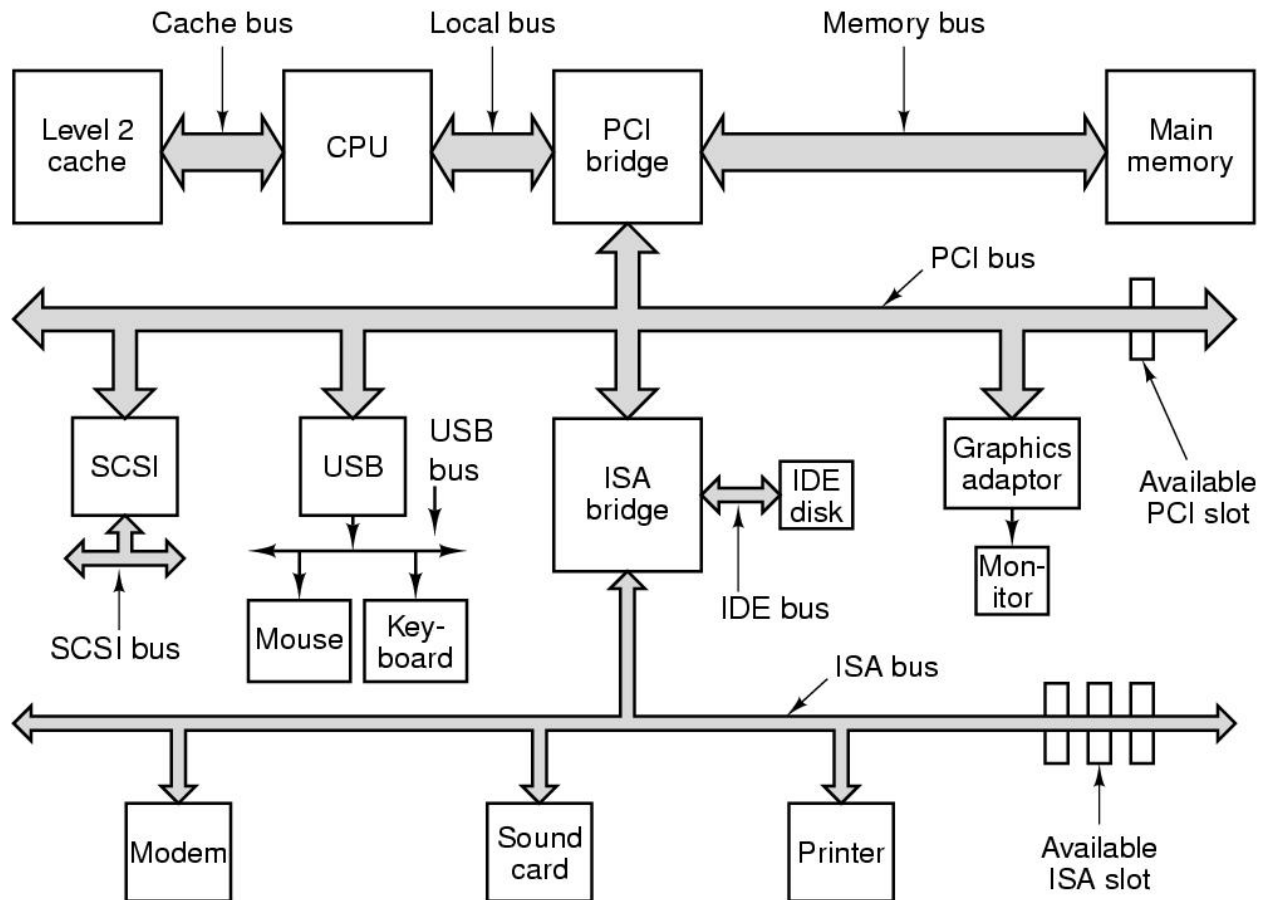
(a)



(b)

Зураг 1-11. (а) О/Г төхөөрөмжийн эхлэх, тасалдах бий болох алхамууд

Шугамууд



Зураг 1-12. Пентиум системийн бүтэц.

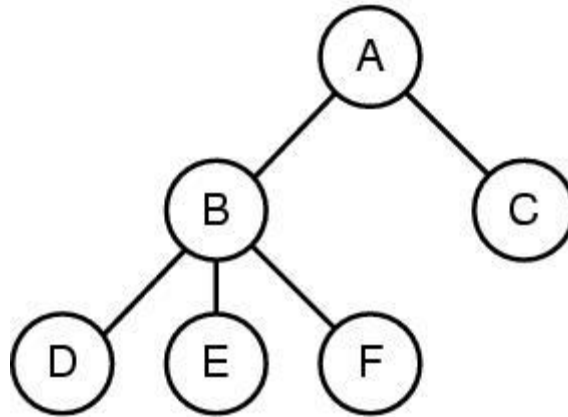
Үйлдлийн системийн төрлүүд

- Мэйнфрейм ү/с
- Серверийн ү/с
- Олон процессорын ү/с
- Персональ компьютерын ү/с
- Зөөврийн ү/с
- Embedded ү/с
- Sensor node ү/с
- Real-time ү/с
- Смарт картын ү/с

Үйлдлийн системийн тухай

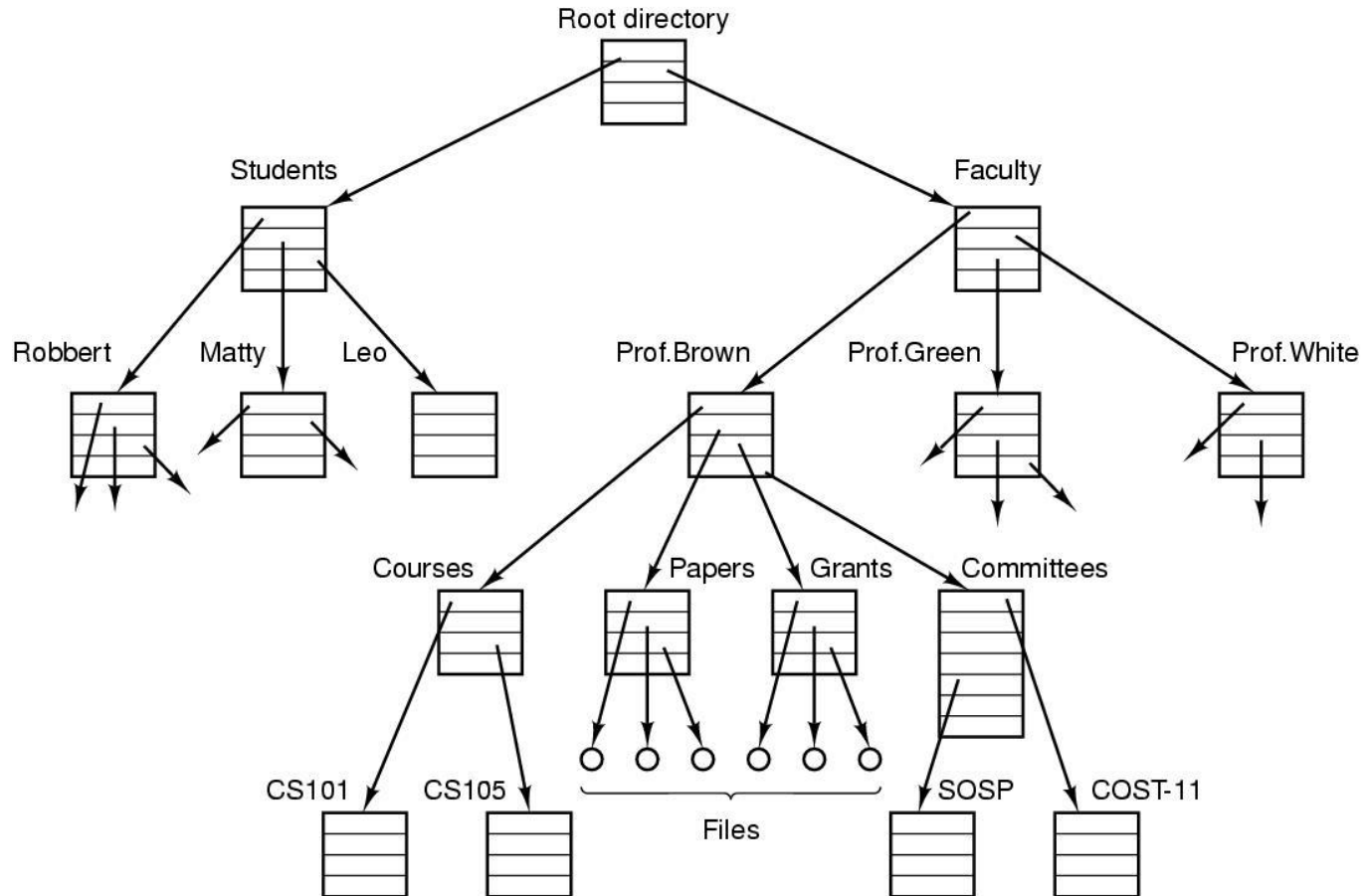
- Процессууд
- Хаягийн орон зай
- Файлууд
- Оролт/Гаралт
- Хамгаалалт
- Шел
- Хөгжил . . .
 - Их хэмжээний санах ой
 - Техник хангамжийн хамгаалалт
 - Дикснүүд
 - Өргөтгөсөн санах ой

Процессууд



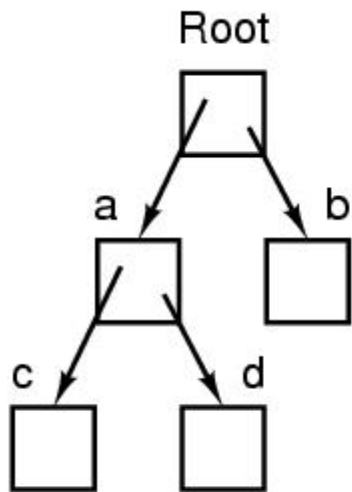
Зураг 1-13. Процессын мод. Процесс А нь В ба С хоёр хүү процессыг үүсгэсэн. Процесс В нь D, E ба F 3 процессыг үүсгэсэн.

Файлууд (1)

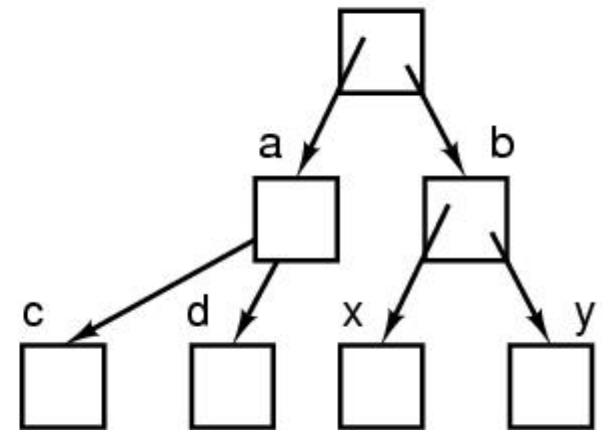
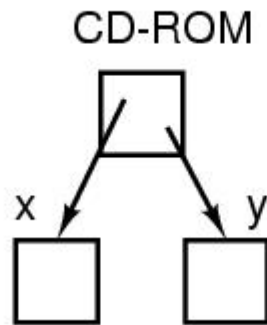


Зураг 1-14. Их сургуулийн тэнхимд зориулсан файл систем.

Файлууд (2)



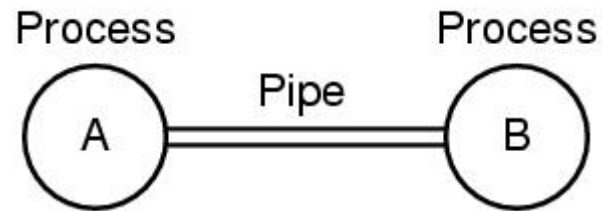
(a)



(b)

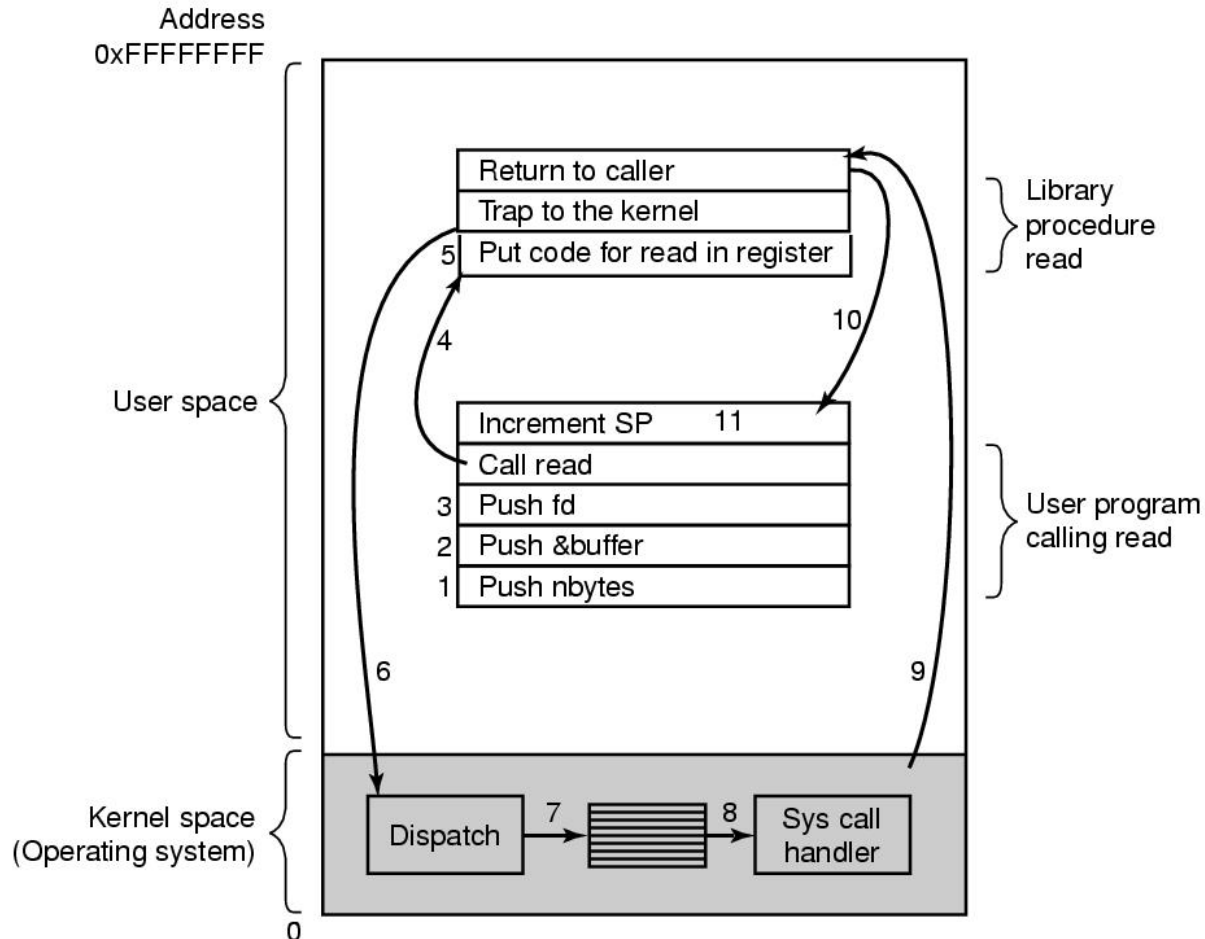
Зураг 1-15. (a) Суурилуулах өмнө, CD-ROM дахь файлуудад хандах боломжгүй (b) Суурилуулсаны дараа, файлын шаталсан хэсэг болно..

Файлууд (3)



Зураг 1-16. Pipe –аар холбогдсон 2 процесс

Системийн дуудлагууд



Зураг 1-17. Системийн дуудлага хийгддэг 11 алхам `read(fd, buffer, nbytes)`.

Процессын удирдлагад зориулагдсан системийн дуудлагууд

Process management

Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status

Зураг 1-18. POSIX системийн үндсэн дуудлагууд.

Файлын удирдлагад зориулагдсан системийн дуудлагууд(1)

File management

Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing, or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information

Зураг 1-18. POSIX системийн үндсэн дуудлагууд.

Файлын удирдлагад зориулагдсан системийн дуудлагууд (2)

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Зураг 1-18. POSIX системийн үндсэн дуудлагууд.

Бусад системийн дуудлагууд

Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970

Зураг 1-18. POSIX системийн үндсэн дуудлагууд.

ЭНГИЙН ШЕЛ

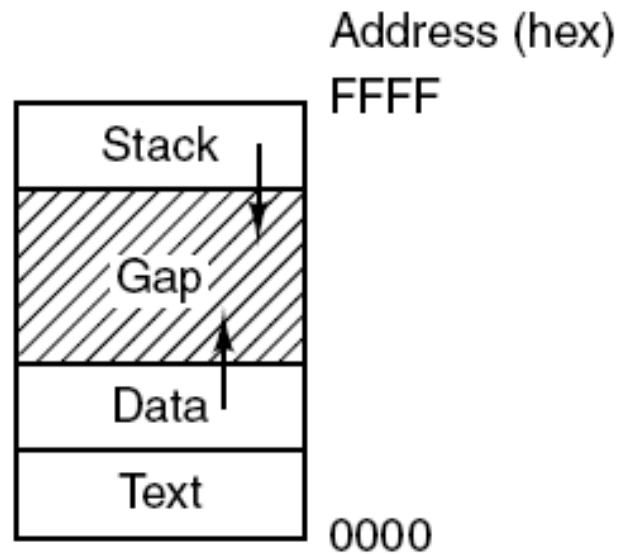
```
#define TRUE 1

while (TRUE) {
    type_prompt( );
    read_command(command, parameters);

    if (fork() != 0) {
        /* Parent code. */
        waitpid(-1, &status, 0);
    } else {
        /* Child code. */
        execve(command, parameters, 0);
    }
}
```

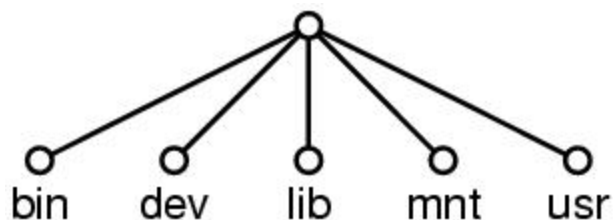
Зураг 1-19. A stripped-down shell.

Санах ойн ерөнхий бүтэц

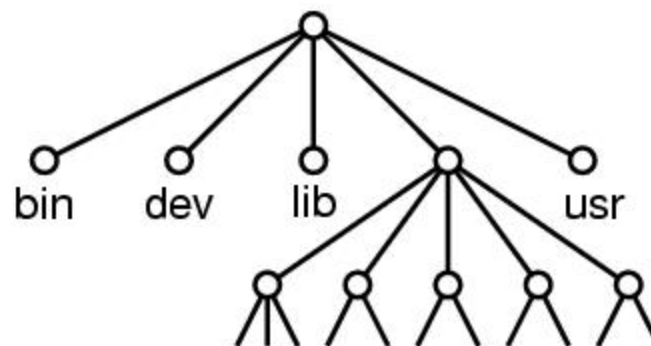


Зураг 1-20. Процессууд 3 сегменттэй байдаг:
текст, өгөгдөл, ба стек.

Суурилуулалт



(a)



(b)

Зураг 1-22. (a) Өргөжүүлэхийн өмнөх файл систем.
(b) Өргөжүүлсэний дараах файл систем.

Windows –H Win32 API

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Зураг 1-23. Win32 API дуудлагууд.

Үйлдлийн системийн бүтэц

Нэгдмэл системүүд – үндсэн бүтэц:

- A main program that invokes the requested service procedure.
- A set of service procedures that carry out the system calls.
- A set of utility procedures that help the service procedures.

Нэгдмэл системүүд

```
#define FALSE 0
#define TRUE 1
#define N      2          /* number of processes */

int turn;                /* whose turn is it? */
int interested[N];      /* all values initially 0 (FALSE) */

void enter_region(int process); /* process is 0 or 1 */
{
    int other;          /* number of the other process */

    other = 1 - process; /* the opposite of process */
    interested[process] = TRUE; /* show that you are interested */
    turn = process; /* set flag */
    while (turn == process && interested[other] == TRUE) /* null statement */ ;
}

void leave_region(int process) /* process: who is leaving */
{
    interested[process] = FALSE; /* indicate departure from critical region */
}
```

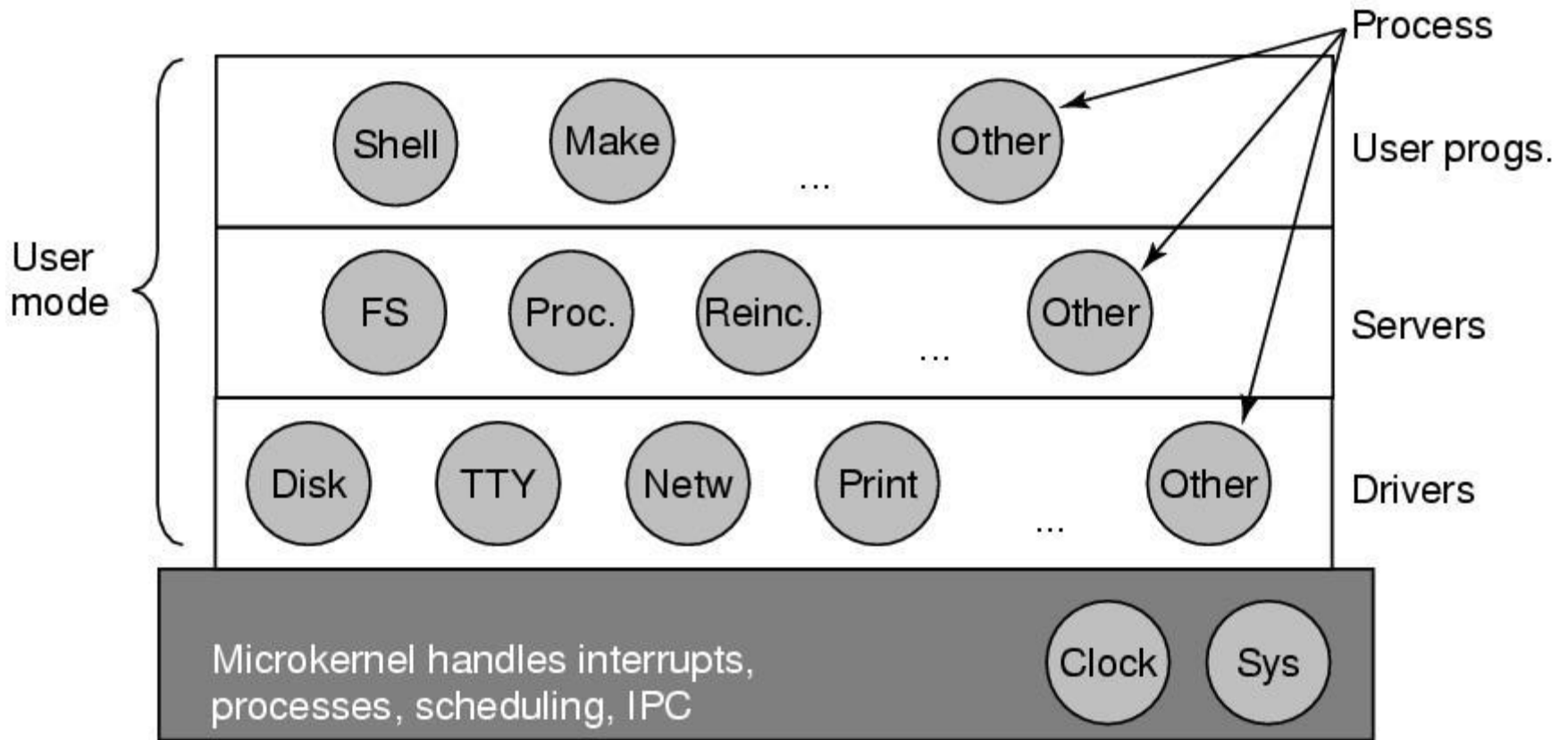
Зураг 1-24. Нэгдмэл системүүдэд зориулагдсан энгийн бүтэцлэгдсэн загвар.

Давхаргат системүүд

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

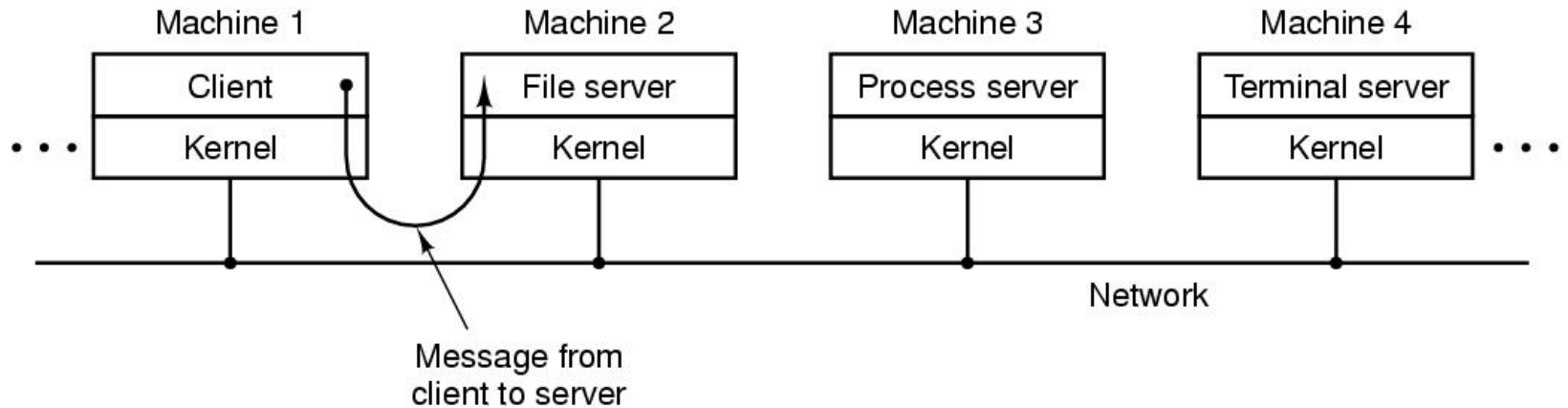
Зураг 1-25. Үйлдлийн системийн бүтэц.

Микроцөм



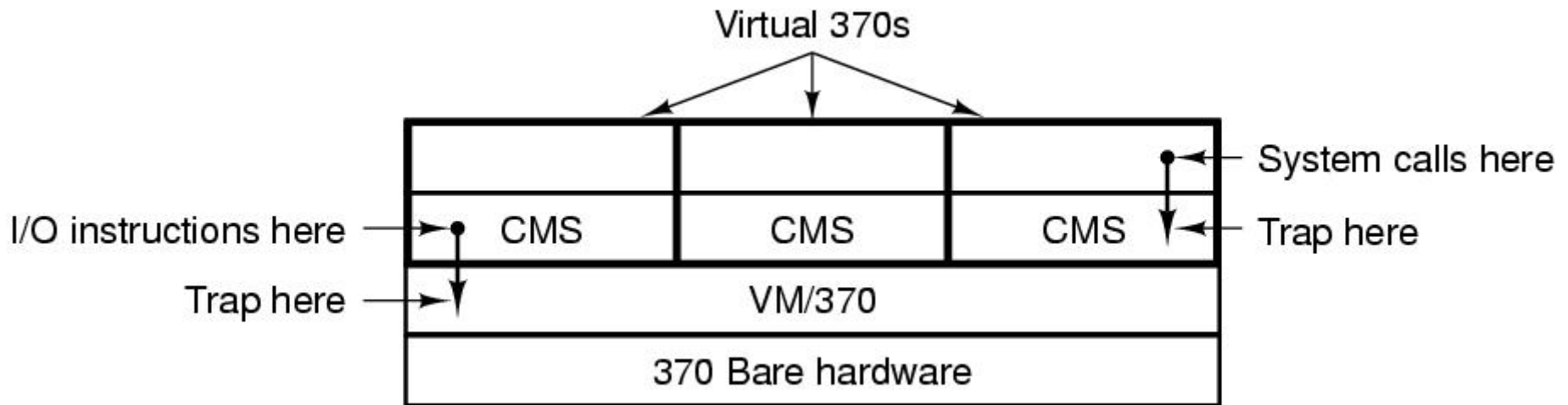
Зураг 1-26. MINIX 3 системийн бүтэц.

Клиент –Сервер загвар



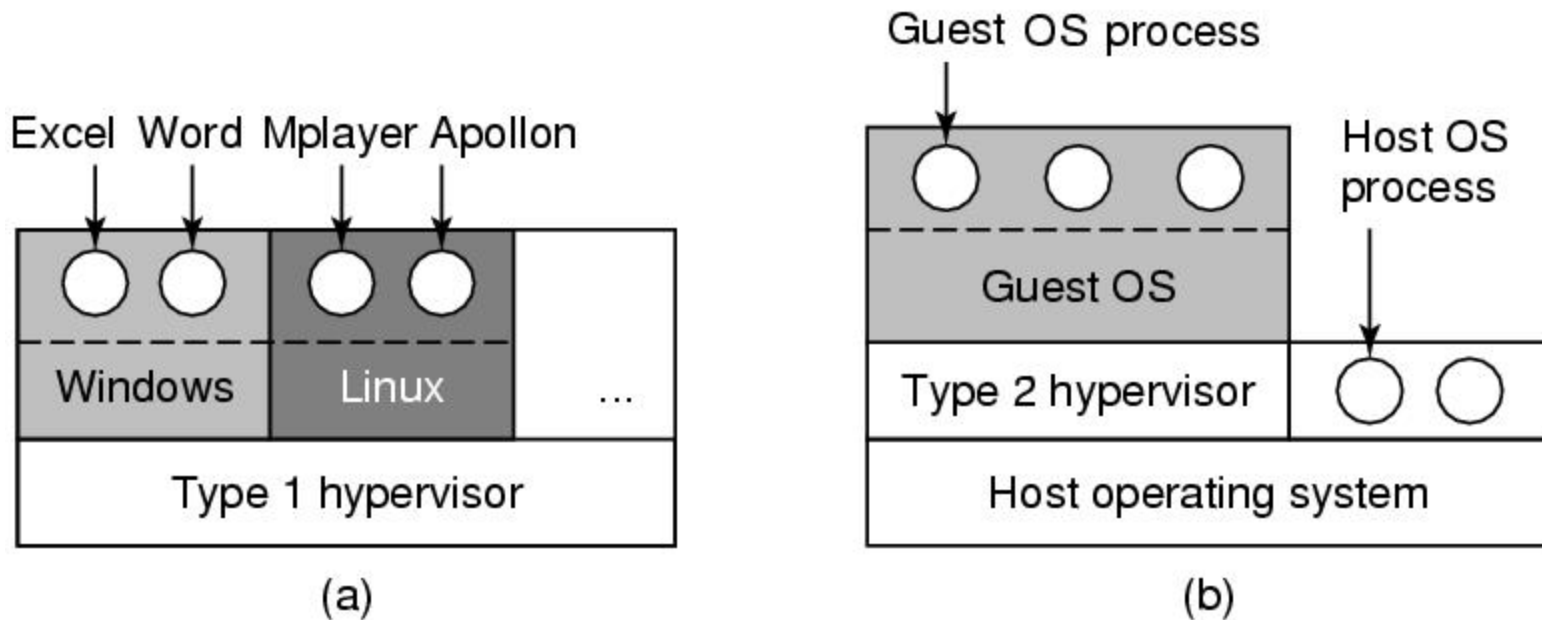
Зураг 1-27. Сүлжээний клиент –сервер загвар.

Виртуаль машинууд (1)



Зураг 1-28. CMS -тэй VM/370 –н бүтэц.

Виртуаль машинууд (2)

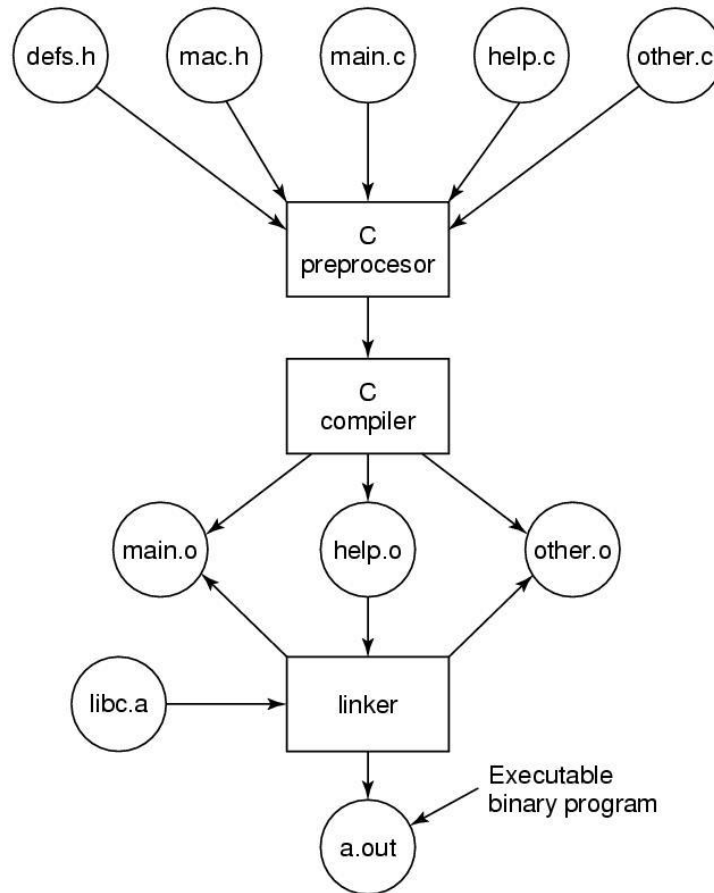


Зураг 1-29. (a) 1 –р төрлийн хайпервизор. (b) 2 –р төрлийн хайпервизор.

Программчлалын C хэл

- Программчлалын C хэл
- Толгой файлууд
- Их хэмжээтэй программчлалын төслүүд
- Ажлын (ажиллах үеийн) загвар

Ажиллах үеийн загвар



Зураг 1-30. C хэлийг хөрвүүлэх процесс