

processor را کنار گذاری کرد و multiprocessor system ایجاد کرد (فرازهای کارایی) الف

میگناریم. و هر چند از آنها المقاره میکنیم. (دینون سستی های Parallel system معنادله روش اند)

النقاره ازایی نوع سیم ها و تکریج را در T و throughput

(صني) م تواند ببرنامه های سستی را اجرا کند و ↑ reliability می سود

بعنی آگر دے پرنسپل خوب نہیں ہے ماضی میں
برائی بعینہ پرنسپل ایکارنسسور

Asymmetric \uparrow will lead to multiplets

Symmetric

کاری کنند .
در این نوع سیستم کامپیوترهای محیا را با کابل های
سیند بینهم وصل می کنند و می توانند بین فضای storage خارجی دسترسی بالند .
و همان از سیستم storage-area network (SAN) می باشد .

و Symmetric (نوع متساوی) و Asymmetric (نوع مختلف) multiprocessor (جهاز متعدد المعالجات)

• IRS ~~is~~ ASymmetric
• IRS Not Standby IRS, IRS \leq e ASymmetric

e Asymmetric clustering

و Symmetric applications اسے مختلف امور کی کتنے داری نورا کی جائے اسی کو applications کا monitor (monitor) کرنے پر مدد ملے گے۔

وَمِنْهُمْ مَنْ يَرْكِعُ وَكَثِيرٌ مِّنْهُمْ لَا يَعْلَمُونَ

HPC  The performance of a cluster is determined by its configuration. DLM is used.

ویرایشی (ویرایشی) با کد و تعامل
برای برنامه‌نویسان (Programmers) که در system call های کرنل
که برای برنامه‌نویسان (Programmers) استفاده می‌شوند.
که رفاقت به کرنل (Kernel) می‌شوند.

برای دسترسی به API ها که در system call های کرنل می‌باشد
برای دسترسی به API های کرنل از طریق API های کرنل می‌باشد که در سیستم کال ها را
برای برنامه‌نویسان (Programmers) ایجاد کرده است. این برنامه‌ها API های کرنل را
در صورتی که برای برنامه‌نویسان (Programmers) استفاده می‌شوند، از طریق API های کرنل
و از سرویس‌ها (Services) استفاده کنند.

سؤال ۱

بررسی میگیریم که با کد \leftarrow برخاطر بوت سسنه بر سیستم چی گذشت که و قی کامپیوکت (روشی میگذرد) با این لذت OS خاصیت بارگذاری نداشت که میتواند با این روشی میگذرد \rightarrow system boot

۲

جهت چندهای اجرا سود \rightarrow راسیون مسقف از هارد دخیل شده این برنامه کوچک و داخل UEFI و امنیات رسیدن از BIOS دارد و بخطاب حلیم که دارد نفع دارد از

BIOS و سسنه را بوت میکند

UEFI و نیز سسنه را تو فتح میکند و دلتورات این این دلتورات هستند که اجرا میگردند (تو RAM (وریون) CPU تا اجرا میگردند که بعد از آن کامپیوکت

GRUB و بوت لوار رسید را لوار کند مثل UEFI

و disk kernel از حینی انتقال اجراهای اینکه VEFI از GRUB

که مختلف option هایی مختلف برای Kernel دارد

و GRUB در تواند حالت بوت را تعین کند که مثلاً Single یا Dual باشد

سؤال ۲

درست است. چون مایل تواند باشد (عویض) صنسر که صانعی در آن دعا کر

به process هایی یوزر و احرازی سیگنال را (هم زمان کارکردن)

بنابراین وظایف برنامه نویس است که این

process هایی که ایجاد کند تا توانند هم زمان از عویض صنسر که اینکه

اما در این Message passing و این ماهیت وظایف OS است و برای برنامه نویس را دارد است.

کامپیوٹر

نادرست است

هر بروسٹ کی ملکوں

PCB بازی رہنچا اور PCB کے حفاظت نہ اسے گواہی نہ ائے
لہ PCB وہیں کہ دستی پہلاں PCB پر اسے
(Kernel memory) کے زیر ڈھونڈیں گے stack (پلٹ)

لیست مجازی "Virtualization"

سهم اقدار

hardware \rightarrow virtualization
که مربوط به کامپیوتد واحد است که داده اجرازه پر را که کامپیوتد

disk \rightarrow memory \rightarrow CPU \rightarrow کامپیوتد باس (network interface)

ایجادی، کنیع \rightarrow یعنی مرتکب نیزی تعداد زیاری بیناط اتفاقه کنیع

به مکانیکی تکنولوژی، این تصور نیزی کاربر و برنامه ها و process ایجاد می شود

که تعداد کامپیوتد دارند از آن کامپیوتد جدا با اتصالات مسئول و ریسورس های جداگانه اتفاقه می کند (نداخنی باقی ندارند)

برای پیش سفت افزار کوچک تر برای (کلاینت) در چابات (ایم) و بزرگ کامپیوتد

کامل (با RAM و CPU و memory) دارند و اینها متوسطه ای

قفسه های سرور که پیش از RAM و CPU و memory دور دارند کلاینت اخراجی در واقعیت

دارند و کلاینت فردی کند که LP در اختیارش است

در پیش صنایع نیزی مور - در واقعیت صنایع بزرگ دور دارند که کلاینت های

(کامپیوتد های قوی آمریکا) مختلف تفاسیم مور اما

کلاینت صریح این قفسه های مور

این تفاسیم سفت افزار و ریسورس ها مدل کلای است که ۰۵ ها انجام می دهند

مدل virtual machine که از منابع می سسخ (کامپیوتد) اتفاقه می کند که جهت

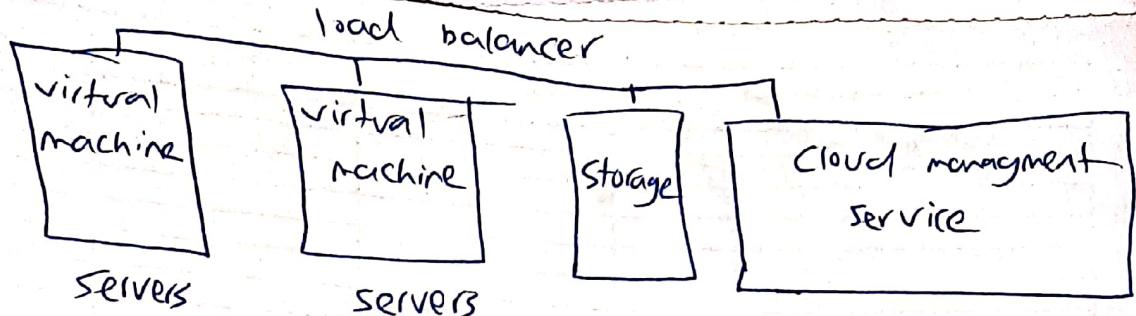
۰۵ را سازد . در حالی که ۰۵ هایی عازی دارند از مدل افزار واحد استفاده

می کنند اما صریح این می تواند اثمار \rightarrow جداگانه و به علاوه

اخراجی دارد که

در سیستم cloud ما، اسما تکی که cloud در اختیار گذارد

را مدل می سرویس می نمیند .



جون سردرهاي مختلف درجه بايد بتوسيع کار را نه و اينجا منابع

و بنابری ۴۵٪ مبالغ فقہاً پیدا نہیں کرے۔
 (صلیل کاری کے ۵۰ درصد حاصلی ساتھ) مکمل

برای این کار CPU می تواند عبارت DMA را اخراج کند در اینجا درستورات و initiate و فرمان در فرستاده از CPU از طبق device register (regiser) های خارجی صل می شود (IO ports) و فرستاره بی لور.

مثلاً این لور ک تواند register ها اهدایی نویسند \leftarrow اقدام لازم را انجام دهد \rightarrow device بتواند register ها بنویسد \leftarrow device . اما این کار را با interrupt انجام داده است.

اسے، آزمیز احتمالی سور
فی ترانس کل ہے اسی سیستم میں memory bus (bus cycle) کہا جاتا ہے

هم device و CPU memory بفروغ همان دسترسی داشته باشد
 و memory بفروگارانه سعی می‌سوزد بقدرت دارای سور ← سرکار CPU
 چون DMA معنی بالرد نمودن باری خواهد بجهت دسترسی داشته باشد. صار
 که زیرا که راهها را فن Cache زننده باشند و memory سخن در ←
 تداخلی باشد DMA بینداخته کند
 اما زیرا که بتواند از فهرست هردو استفاده کند و CPU قابل کنترل ای که
 توسط device احراست که ایام سور

اگر رابط حافظه (memory interface) وجود داشته باشد سند اقدامی
باید دارای یعنی رضامس است ها و صور خواهد داشت.

خطوار آن و اولویتی که به Processor می‌خورد نسبت به DMA نیز ندارد اما
اگر رهاکی نداشته باشد I/O و دیوار (Starving) یا کرسنی نشود اما حتی اگر
DMA نمی‌تواند اولویت داشته باشد Processor و همچنان فرستادهای باید دستکاری
باشد. جون I/O و پنهانی باید کمتری نیاز داشت memory

اما صفوای (soft) از رابط باید حافظه و صور دار و بردازندۀ هایی که برفور منعی باشند
دارند لزی استفاده شوند.

اگر کنترلر cache و DMA نیستند و cache حتی نیازی به ردیابی ندارد
حتی اگر کنترلر ردیابی کنند و overhead ها به ریلی باید باید سنا
احتیاط memory و Cache کوچک شود.

پرسوکنترلرها، ایندیکاتور instruction از قدری که حافظه جداگانه مبنی نیز
فقط و دستگاه پیدا می‌کنند که اجازه می‌دهند درین احیان DMA و همیار
انجام سود. که انجام حافظه ای کم باشد رابط مسئول دارند ←
(اجازه ردیابی به تعداد زیادی داره بدن تدخل داشتن) DMA

* حتی با داشتن رابط حافظه واحد و حد اکثر بینایی باید صفوای (soft) از حد اکثر بینایی
باید نداشته باشد که اینستار (I) می‌سوزد. صندوق باید Fetch کردن و حتی پافد کوچک که
می‌باشد (وریسدر) نسبت به میانگین لور دارد و memory و لذت بر اجازه می‌دهد
که حقیقتی هم buffer داره توسیع نمی‌رسد (لذت بر اینستار) اینستار انجام سود
و باشد می‌مانند (برای) دارند که درین دستگاهی دارند و instruction ها و صور
دارند.

6) modular kernel و monotonic (جزءیت)

(CPU scheduling)

6) monotonic (جذب اسفاره از زیرگ)

سی از فایلهای بزرگ اینها، قدرتمند نیست که اینها را در مدتی از میانه اینها خود را بازگرداند.

فایلهای ریس این که در این نوع اینها، system call OS از میانه اینها داده است.

که در فناوری آدرس واحد اینها میتوانند میانه اینها را بازگردانند.

و همچنان static binary file میتوانند اینها را بازگردانند.

این نوع سیستم کار اینکه kernel و UNIX 6) اینها میتوانند اینها را بازگردانند.

که در فناوری آدرس واحد اینها میتوانند اینها را بازگردانند.

این روش سیستم کار اینها میتوانند اینها را بازگردانند.

این روش سیستم کار اینها میتوانند اینها را بازگردانند.

خوب سونه (fail-safe) ← کل سیستم که میتواند از نویسیده.

آن بعده کوچک سیستم که میتوانند اینها را بازگردانند.

رویکردن سیستم اینها است.

6) modular kernel و OS

فوانی اینها از لورکنده ← Kernel نیاز نیست که 6) اینها را در زمان boot و را در زمان اینها را در زمان زمان بعد سیستم که میتوانند اینها را بازگردانند.

آن بعده ماژول جایگزینی نیازی به کامپیل کردن ندارد.

کامپیل و ماژول 6) قبل لورکنده نیازی به اینجا فایل نیست.

لورکنده حین رویکردن تسلیم نمایند.

افزونه کرن feature و تغییرات اینها را در اینها.

6) مکانیزم اینها

برای مسائل اینها 6) حین هر چند module اینها را بازگردانند، اینها را نمایند.

از تواند ماژول که باشند crash - OS میتوانند.

برای همین دلیل سیستم عاملها 6) میتوانند با زمانی اینها را لورکنده کنند.

برای همین دلیل سیستم عاملها 6) میتوانند با زمانی اینها را لورکنده کنند.

۳) کرنلی (kernel) می‌تواند سفت‌ware باشد جون procedure و module را به کرنل ارجاع دهد

در سیستم از تقاضه کاری (QoS) برای performance بود است (جون)

لیکن راهنمایی دار و بسیار نیزگاه monolithic های اسر که آنها کار نمی‌کنند

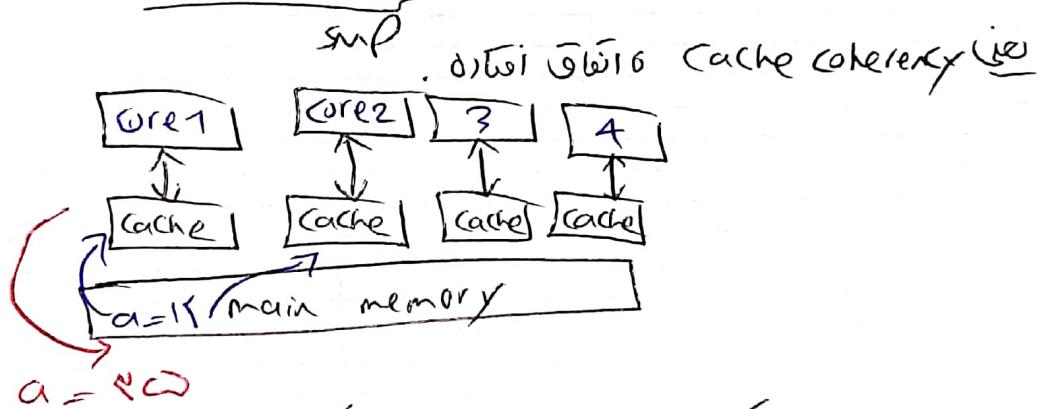
سازمانی fail و kernel fail سیستم کلی خوب نیست

از تقاضه انتظاف پذیری هم می‌باشد (modular) بود است (جون) نیاز به کامپیویلین وجود

از تقاضه انتظاف پذیری هم می‌باشد (modular) بود است (جون) نیاز به کامپیویلین وجود

بررسی

آنچه متواتر می‌شوند دریم و فور داشته باشیم

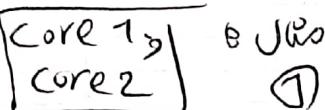


آنچه اتفاق زمایی می‌افتد که باید تغییر پسندی کلی از مقدار را کوئی و فور

دانه باشند و باید از core های دارای این مقدار کنار بیندازند

در این حالت باید از مقداری دار که قدری در آن است

و عددی کلی از متغیر a دارند در عساوه.



مقدار این متغیر از memory خوانده شد و در این صورت

مقدار متساوی از a را دارند.

a = 1 عددی مثبت است

Core 2 که ذخیره کند و a که نگذارد صدای دیند

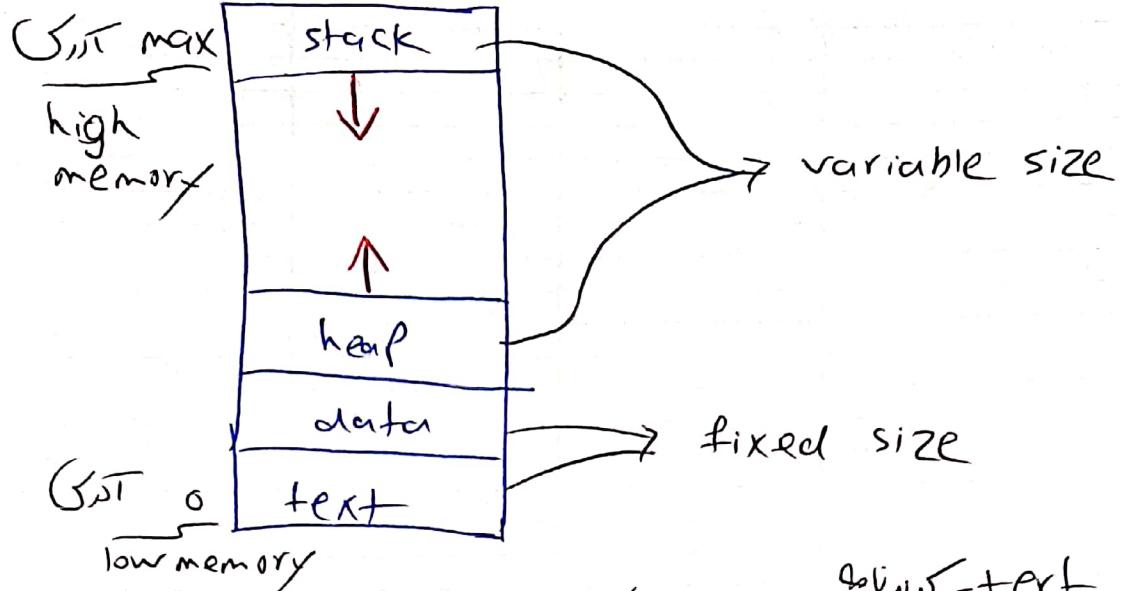
بک دستور load از متغیر احمد کند هم زمان و دستور خواندن با a مقدار جدید را بخواهد.

و فوراً a بخواهد + به این مطالعه عدم رسانی Cache هایی کوئی نگوئند.

برای این درایفه ایلی ب

فال

الف



نامنکر text ①

میتوانیم بسوار داشت و heap ②

ذخیره کرد و local است و ممکن است ذخیره شود ③

با استفاده از function ها می توانیم اینها را با function های دیگر ترکیب کنیم

int y; \leftarrow ممکن است uninitialized data باشد ④
int m=10; \leftarrow initialized data

که CPU را به process می بندیم و با آن می توانیم context switching را انجام داد

process به process از خواهد بود و سری سویغ کرد.

منفرد از context یعنی اطلاعات است که در آن دارای محتواست

و در حال اجراش را با پردازنده P_0 می بیند و PCB را می بیند

اطلاعاتی که در آن دارد را با P_1 پرداخت می کند (درایفه ایلی)

و مخفی بال رجیسترهاي از CPU را در خواهد بود

برای P_0 در حال اجرا P_1 را در خواهد بود

برای P_1 در حال اجرا P_0 را در خواهد بود

Save state into PCB

برای P_1 در حال اجرا P_0 را در خواهد بود

Reload state from PCB

با افزایش complexity
هم اقدامات بیشتر می‌باشد.
Context switching ← زمان عبوری PCB و OS

(waiting ~ running) خود را دوست کند (از
با بدلی)

بنواده این کار را نماید.

process scheduler function

(حالات اجتیاری)

I/O (input & output) دالله ایل
حالات احیانی process نیز بدین
باشد از حالت اجتیاری استوار نیز

process scheduling بین احیانی و دخواست نهایی I/O

استفار برای دریافت یا حذف کردن آن را با احیانی process انجام می‌کند.

عمل

حین اول والد اجتیاری شود و بعد فرزند

(الف) حین اول والد اجتیاری را تغییر داده $a = 41$ ← مقدار صفتی را تغییر داده

(ب) خنده کن تواند بخاند حین والد قبلی فایل را close کرده

2. child هم راضی سده معنی wait، parent هم راضی سده

مخفی و قبل از آنکه فرزند هام شود (درین جای قبلاً از لروع گشته) و والد اجتیاری

سده و بین راهه و wait هم راضی سده ← هم راضی عدم معنی

برای حل مسئله ← با `wait()` تابع را فراخوانی کنیم

int status=0

if (ref > 0) { والد

wait(status);}

5

نوعاً

int[] سعی در اینجا تا \leq کسر (کسر \leq) \leftarrow pipe (fd) ①

اس راچ کیور و اگر ساخت یاں موقعت امن بالہ ہے ~~مکان~~ ایسا گا

ساحل file descriptor حبیب لور سایری بایبلان القداد لوند.

۱۷

الآن بانها واجه dup1 و dup2 (ج) ②

نوزر مسٹر فائیل دکٹر ایک file descriptor از مکانیزم استعارہ سے کر کے دھارہ گذرا جائے گا اور اسے ایک descriptor کی طرح استعارہ میں لے کر تو سے

وهي تفتح الملفات وتحصل على file descriptor.

اگر کیوں صرفیت امنیت بالے ہے تو اسے file descriptor کہا جاتا ہے۔

اٹھی بے استھانوں کا copy اسٹھان

و هر دو file descriptions که باز نمایش داده می شوند اینها را می توانند

int dup2(int oldfd, int newfd)

oldfd → old file descriptor

newfd → new ~ ~

كَوْسَهُ الْمُرْبَقِيَّةِ، إِسْتَارِهِ، وَالْمُرْبَقِيَّةِ

اولیٰ تھوڑے made ہے جسے ~~کہاں~~ کہاں تھے اور $\text{dmp2}(\text{fd}(1), 2)$ تھے۔

```

graph TD
    stdio[stdio] --> stdioout[stdioout]
    stdio --> stderr[stderr]
    stdioout --> stdio
    stdioout --> stderr
    stderr --> stdioout

```

اگر اورڈری) "نولٹن رافل پائیلائیں رخ رار ← توی استار چاہ کن

٦) $f1(7)$ معنی آن چیزی است که $\text{dup2}(f1(7), 2)$ می‌بینیم

و (سُوكِنْجِي) fd[7] - file descriptor (stderr) سُوكِنْجِي

وَالْمُنْعَلِّمُ لَهُ فَلَدَنْتَهُ سَعَى وَالْمُنْعَلِّمُ لَهُ فَلَدَنْتَهُ سَعَى

✓ `printf` چیزی را که در آن قسم از کوچک باینی \leftarrow `dup2(fd[1], STDOUT_FILENO)`
خوبی خوبی کنم و پس از کند باید داشت و داخل $(fd[1])$ را می‌نویسند
کوچک کردن = standard output

بروسه را به صورت (فایل) از پایداران کر می‌توانیم تغییر داده.
 int execv (const char *path, char * const argv[]) ← execv(Command, args)

`path = argv[1] ← argv[1] از آرگومان ها که میشه برای یادوگار جدید در نظر گرفت`

کو فنا فعلی (exec) کا Command (Goto) ← exec(Command),
کرنے کی بنا پر (argv1) کو فیکٹری میں اگر کوئی فہرست (list) ہے تو اسے
argn) کے لیے اگر کوئی فہرست (list) ہے تو اسے argn) کے لیے اگر کوئی فہرست (list) ہے تو اسے

- This makePipeline() function takes arguments

$$\text{d}v\rho_2(f_{\mathcal{A}(U)}, \omega)$$

~ fd[0] & pip is reading file descriptor $\xrightarrow{\text{من کوئی duplicate کرنا کریں}}$
 ~ input is command $\xrightarrow{\text{ایک کمینڈ کو کوئی دوسرے طرف کرنا کریں}}$ standard input file descriptor
 $\xleftarrow{\text{ایک کمینڈ کو کوئی دوسرے طرف کرنا کریں}}$

is it new \leftarrow grep new
is it Stalin \leftarrow command

← قیمت خود را stelout نویسید (در کتاب لری) [۱۵]

رسالة dup2(fd[1], STDOUT_FILENO)؛
كرر.

کریں۔ exit() وان مسٹری لور-اُفزنان د اچرسونز د wait(NULL) کال کندہ

صفحه توافقی دارای

۱) می خواهیم با اعمال یکسانی دستور (new | grep new) را اجرا کنیم
دستور اویی کا است که خروجی آن ۲ باینری ترمینال های دستور
دستور دویم هم است که وردی هم دارد و new که باشد از خروجی
ترمینال گرفته شود

۲) اول می توانیم pipeline می بازیم و سپس کیه را بازی زفایل کردن کردی که
میگرداند

۳) call کرنیم و سپس آنرا بازی زفایل کنیم

۴) اگر سیمه fork و ۱-بور می باشد ساخت را بازی خوب نمیتوانیم
میباید بور را در فریلنده و الد نمیتوانیم
صفدر ~ فرزند ~

۵) وقتی در فرزند میخواهیم دستور اویی را اجرا کنیم و حون در آن
دستور باید بنویسیم (خروجی خواهد بود) ← در حواندن pipeline را می بینیم و
با تابع dup2 و کمی از ند نوشتی یکسانی میگیریم و میگیریم تیکی

۶) را تکی ترمینال های را باز کنیم و میگیریم برای افعی را با Command اویی باز کنیم

۷) در والد را کمی بروزی می بازیم و اگر در آن بروزی کمی در فرزند بروزی ←
میگیریم لرانی اعمال دستور داشتیم ← حون باید واردی بسیج از ترمینال ها در حواندن
PIPE را می دهیم به std::in و دستور داشم را اجرا کنیم

اگر دستورها اجرا نشون ← بیگانه command no such command

در والد را کمی بروزی داشتم و قبیر کنیم تا اعمالی بروزی های اویل و داشم
کام کور و دستورات pipeline را می بینیم ←

86