

① Diferențe formale dintre W și U: W - d: \dint\|dint\|dint\| fisier; U: dint/dintm, Rath W: w; u;
U: L: L; L. Rath operator; fisier de comentari: fisier arg1...argm W: %D %I %X; U: \$0 \$1...
Opțiuni W: /optione; U: -optione; Fisier txt: W: to tm-; U: -tm-; ② Comanda Unix „commando”
opțiuni expresii fisier”; commando = numele comentari; opțiuni = este specificată de abrice printro singură literă. Un
grup de opțiuni este precedat de „-”; expresii = sunt șiruri de caractere și commando resp; fisier = repr. unul
sau mai multe fisier specificate relativ sau absolut, e pot utiliza și specificări generale. În mod implicit comentari
u le sunt asociate 3 fisier: fis standard de intrare, - de ieșire și - de eroare; la terminarea executie aricare
comentari în sistem se returnează un cod de retur (m. integ 0, 255)

③ Expresii regulate: este o machetă prin care se indică sintaxa pe care un anumit șir tb. s-o respecte. Stringurile
care se potriveșc sunt stringuri care verifică expresia regulată. Construcții de expresii regulate: „c” caracter;
„.” orice caracter, ic c și pierde statutul de caracter special, „^” începutul unei linii, „\$” sf. unei linii;
„\”, „\” începutul și sf. unui cuv., „|” alipirea unei expr. reg., „*” concatenarea unei expr. reg. de mai
multe ori, „+” concatenarea repetată a expr. reg și cel puțin odată; „?” poate să apară o dată sau nu odată,
„|” sau „|” sau „|”, /șir/ urm. linie față de cea curentă care conține șir, ?șir? precedentă...

④ Specificarea fis U: pt. denumirea unui fis nu se pot fol bi mai, nu, o fre, „.” și „-”, se recomandă evitarea fol. unor
caractere speciale: <> | ^ % * ? ! ~ / . Fisierul al căror nume încep cu „.” sunt fis speciale în linux cu rol de referire
a unor inf. utile din so. Caracterul „.” are rolul de a înlocui orice succesiune de caract iar „?” substituie un sg caract.

⑤ Filtru U: filtru = prog. care modifică sau transf. intrarea sa. Filtrul citește rândul pe rând fis standard de intrare
și afișează la ieșirea standard rândurile modificate. Ex filtre U: grep, sed, sort, tr, cut, head, tail, paste.

⑥ Interpretare Shell: sh, ksh, csh, tsh, bash; W: bat, Powershell. Alg. după care funcționează shell-ul Bourne
cât timp (nu s-a închis sesiunea) o fișecă prompter; citește linia de comandă; Dacă (linia se termină cu)
aruncă creato un proces și-l dă spre executie comandă; Nu așteaptă ca executie să se termine; altfel
crează un proces... Limbajul Shell are 13 cuv. rezervate: if, then, else, elif, fi, case, in, esac, for, while, until, done, da.

„|” legare pipe, && legare and true, || legare or false, „;” separator/terminator comandă, {} grupuri de comentari;
* înlocuiește orice șir de caract, ? înloc un sg caract, * înloc cu orice caract. din șir; „\$var” valoarea variabilei
shell var, „commando” ieșirea comentari comandă, \ evitarea caracterului următor, „...” evitarea caracterelor din... cu
excepția „...” evitarea caract din... cu excepția „...”

⑦ Gruparea proceselor U: cam 1 && cam 2 - se exec cam 1; dacă exec. ei s-a terminat cu succes se exec cam 2, cam 1 || cam 2 - invers
„&” indică lansarea unei comentari în background, „;” separator două comentari care se vor executa succesiv, „()” - nu modifică
mediul exterior, „{}” - invers ⑧ Redirecționarea fis standard: Red intrării std. se poate face în 2 moduri: 1) comandă < fisier
intrarea std a comentari nu va fi de la tastatură ci de la fis specificat; 2) comandă: << cuvânt | suce de linie | cuvânt
intrarea comentari repr. aceste linii text, Redirect. ieșirii std se poate face în 2 moduri: 1) \$camandă > fisier - rezultatul
comentari în fisier; dacă fis specificat & atunci e creat automat, altfel, va fi cuprins; 2) comandă > fisier - asemănător
primelor dar cu două fisierul și se va alina în el, nu se va suprascrie. Legarea în pipe a comentariilor &
camandă 1 | camandă 2 => ieșirea standard a primei comentari e intrarea std. celei de-a doua

⑨ Variabile Shell: Definiția unei variabile coincide cu atribuirea unei valori inițiale și se face printro construcție
de forma \$nume=cuvânt. Substituția valorii unei var. shell se poate specifica în mai multe moduri: \$nume; \$nume;
\${nume-cuv}; \${nume=cuv}; \${nume+cuv}; \${nume?cuv?}. Primele două modalități dau ca rezultat variabile;
șirul vid; rezultatul expr. este valoarea variabilei nume, dacă aceasta e definită, altfel e valoarea cuvânt = rez
este val. variab. nume după ce aceasta i se asignează eventual valoarea 0; + dacă variabila nume este definită
at i se asignează val. cuvânt, altfel rămâne în continuare fără valoare; ? rez. expr. este valoarea variabilei nume dacă
aceasta e definită, altfel este afișat mesajul cuvânt. Referința la valoarea unei variabile se face prin numele e precedat
de \$

⑩ Variabile Shell predefinite: variabila de mediu este variabila accesibilă tuturor proceselor fii ale celui proces shell.
HOME, user, logname, shell, mail, PS1 - șirul de caractere al prompterului principal asociat shell-ului; PS2 - prompter sec,
term - tipul de terminal, PATH - a listă de direcție în care shell-ul caută fis corespunzător unei comentari testate, ce a fost
specificată doar prin nume, IPS - cantine separată shell prin cuvinte, acestea sunt: spatie, <TAB> și netru de car;

MailCheck fixează intervalul de timp la care se face controlul de mes. primite. Referința la arg din linia de comandă:
\$camandă arg1, arg2, argm. \$0 desemnează comandă în timp ce argi e desemnat de \$1; \$* toate arg. linie comandă
prive ca un șir; \$@ - , ca succesiune de stringuri; \$! indică arg. ce conțin opțiunile linie de comandă; \$? întoarce
codul de retur; \$! indică PID-ul ultimului proces lansat în background. ⑪ Comentari utile în context shell: sleep, test,
expr, true, false, sleep - are ca efect suspendarea procesului care a lansat comandă pt. un nr. specificat de secunde: sleep n
test - primește ca param o cand. pe care a evaluată (cod de retur); dacă condiția e adevărată codul de retur se fixează
la valoarea 0, altfel la o valoare nenulă. Lansare; test condiție sau !condiție 1. Expr - comandă calculează
valoarea unei expresii aritmetice întregi și tipărește la ieșirea std. valoarea acesteia. Valabile true și false sunt fol
pt a putea descrie cicluri infinite. Terminarea acestor cicluri se va face din intervalul lor prin intermediul opelurilor
exit, break, continue. Specif true întoarce întotdeauna codul de retur 0 iar false 1.

⑫ Structuri de control: if: if listaCam then listaCam elif listaCam then listaCam? else listaCam fi. Se poate vedea
din specificarea sintaxei că în cadrul unei instrucțiuni if pot să apară ori de câte ori perechi elif...then. De asemenea,
înainte de sf. instrucțiuni poate apărea cel mult o dată construcția else... Case: case cuvânt în [cuvânt] [cuvânt] fi
listaCam;; fi + esac; se compară primul cuvânt, cel după care se face selecția, pe rând de la stânga spre dreapta
cu fiecare dintre celelalte cuvinte. For: for nume do listaCam done, for nume in [cuvânt] [cuvânt] do listaCam done
Parcurgerea cuvânt descrie a listă de valori pe care le ia succesiv variabila și pt. fiecare asemenea valoare
se execută comentariile din lista de comentari. While: while listaCam do listaCam done until: until listaCam
do listaCam done. Break: break l m; Continue: continue l m; se referă la părăsirea respectiv reintrarea
în ciclurile declarate prin construcțiile for, while sau until

15) Tratarea liniei de comandă: efectuarea parcurii liniei respective considerând ca și separată caracteristicile altele (spoturi, tab, linie mare). Apoi verifică dacă primul cuvânt din linie identifică o comandă internă shell-ului sau un program localizat în sist. de fisiere. În cazul în care primul cuv. este o comandă internă această este executată intern de către shell, altfel shell-ul caută comanda specificată în toate dir specificate în variabila de mediu PATH. Dacă comanda este găsită, shell-ul creează un nou proces care va executa comanda, așteptând terminarea acestui proces și afișând dacă este cazul eventuala rezultate.

Linia de comandă este descompusă în comenzi elementare, luându-se în considerare separată, 1 și 2. 1) Fiecare comandă elementară este împărțită în cuvinte separate prin caractere albe. 2) Trebuie să se țină seama de variabilele shell. 3) Trebuie să se țină seama de substit. fis. generice. 4) Substit. iesirilor unor comenzi. 5) Se efectuează necesar. 6) Se definește variabilele shell. 7) Se fixează valorile var. \$0, \$1, ..., \$#, \$* fără a se fixa \$?. 8) Se execută comanda. 9) Se stabilește valoarea variabilei \$?. 10) Sist. de fisiere: tipuri de fisiere: Normale (cuv. de adre.), Directori: conține lista de nume și adrese pt fisiere subordonate lui, legături hard, legături simbolice, socketuri FIFO - pipe cu nume, periferice ca main caracter, periferice bloc, mai există 4 entități: Pipe, SMP, CM, semafoare. Principiul structurii arborescente de fisiere este acela că orice fis sau director are un sg. părinte. Autonom, pt fiecare director sau fisier există o sa cale de la rădăcina la directorul curent. Legătură între un fisier sau director și părinte a unui nou leg. naturală.

16) Leg. U, conceptul de mantare 1) Legături hard: pot fi create doar de admin. sistemului; este o intrare într-un director care punctează spre o substructură din sist. de fisiere spre care punctează deja leg. sa naturală; un fisier primitiv efectiv 2 nume; fisierele punctate prin leg. hard apar dublicate; pot fi făcute în înt. aceluiași sist. de fisiere.

2) Legături simbolice: se compartă ca și un subdirector al directorului în care s-a creat intrarea; se creează prin comanda: ln -s cale/instructura/bet/director nume simbolic. 3) Conceptul de mantare: operația de mantare constă în concatenarea unui sist. de fisiere de pe un anumit disc, la un director existent pe sist. de fisiere implicit ex: mantare /opt/umr/3 sist. fisiere director Mantare; opțiunile: -w citirea și scrierea în sistemul mantat; -no citire efectul invers și se face ca comanda: #/etc/umr/mantare din de Mantare

16) Protecția fis. U: dreptul de citire (r-read), de scriere (w-write) - include crearea sublinii, mod. f., etc.; dreptul de execuție: permite lansarea în execuție unui fisier. Drepturile se pot face cu ai comenzi shell chmod q, r, w, o. După lansarea în execuție de către utiliz. a lui P, în mod setuid/setgid acțiunile pe care programul P le poate efectua asupra fisierei sunt cele permise de drepturile proprietarului programului p și ale grupului din care face parte acesta. 17) Legături între procese U: traseul accesului unui proces la un fisier trece prin mai multe nivele: proces, sistem, mod, fisier. Nivelul proces: fiecare proces își întretine o tabelă proprie în care înregistrează toate fisierele lui deschise. Nivelul sistem: întretine o tabelă unică cu toate fis. deschise de către toate procesele din sistem. Nivelul mod este de fapt țara de mod. de pe disc. Nivelul fisier este repr. de blocurile din a fisierei.

18) Funcții de acces la fisiere: print, scan, file, f-open, fclose (f) print (f), fscan (f), fgets (f) - citește a linie întregă de la o succesiune de octeți, fwrite - scrie, fseek - schimbă punctul unde urmează, fsetl - standard C (open, close, read, write, fseek, dup2, fcntl) care repr. punctul de intrare în nucleul Unix și ale căror prototipuri se află în fisierele header <csa.h>. Funcțiile sunt cuprinse în standardul POSIX și repr. orice fisier print-un întreg negativ. Fiecare descriptor este index în tabela de fisiere deschise a procesului. Open întoarce un întreg. În caz de eșec întoarce -1. int open (char * nume, int flag, drepturi). I/O POSIX opeluri: sistem int f, i=open(-), close(i), read, write, lseek.

19) Imag. unui proces U: Este compus din două părți: 1) contextul user cuprinde: 1) țara text - partiunea de instr. marcată a procesului; 2) Datele initiale read/only 3) Datele initiale write/only - acele var ale procesului care primesc valori initiale în țara de compilare 4) Datele inițializate cantine țara ocupată de către restul variabilelor; 5) Heap - țara unde se alocă spațiu pt. var. dinamice. 2) Contextul nucleu cuprinde: 1) starea procesului; 2) pointer la țara user și neg. de mem a procesului; 3) din procesului de memorie; 4) PID, PPID 5) UID; 6) GID (Group ID); 7) EUID; 8) EGID

20) Apelul sist. de lucru cu procese: 1) fork() - din procesul părinte se caprăz imaginea procesului, creându-se un nou proces în prima fază identic cu procesul inițial, rezultat >0 => suntem în părinte, rezultat =0 => suntem în fiu. 2) exec() apelul exec i se furnizează numele unui fisier executabil iar conținutul acestuia se suprapune peste programul procesului curent. 3) exit(): exit (int n) prăvăză terminarea procesului curent și revenirea la procesul părinte în este codul de retur cu care se termină procesul. 4) wait() suspendă execuția progr. părinte la terminarea procesului fiu. 5) waitpid() - pune prog. în așteptare până la apariția unui client eveniment: a) terminarea procesului printr-un sig. pld, b) recepționarea unui semnal de termin. a procesului. curent c) apariția unui semnal cu efect de rezerve din așteptare.

21) Crearea pipe: Conducerea e o cale de leg. care poate fi stabilită între două procese înrudite. Procesul părinte creează un pipe - părintele operează fork() pt a crea un fiu, - fiul include unul din copete (ex. de citire) - părintele include capatul de scriere, - fiul scrie date în pipe folosind descriptorul numas deschis, - părintele citește din pipe din capatul de citire.

22) Open a system: 1) system: creează un nou proces în care cheamă shell, shell cheamă comanda execută apoi se reintoarce controlul la open: FILE * popen (const char * comandă, const char * tip) // tip: "r", "w", int release (FILE * f); deschide un pipe, execută un fork

23) Comunicare prin FIFO: Comunicarea se face între procese care nu pot să prăvăză din procesul care a deschis fisierele. Crearea unui fisier fifo se face prin opelul ft. sistem mkfifo sau. Deschiderea fis. fifo se face cu funcția open(). Citirea read() scriere f. sistem write(); închiderea f. close(); distrugerea unlink(). 24) IPC: Se creează un IPC, nucleu întoarce un handle descriptor al canalului IPC. Pt a permite comunicarea între procese independente este def. de ident. fiecare IPC. În mem. creată, val. chei este transmisă nucleului, iar clientii utiliz. această chei de comun. acan. p. a. și deschide în propriile programe canalele la ipc. În mem. creată val. chei. Drepturile de acces sunt specificate în campul unde mod. și sunt sunt similare cu drept. cc. U. (r, w, x). 25) IPC sistem: crea sau deschide canale de comunicare (shmget, msgget, semget), canal (shm, msgctl, semctl).

26) Mem. partajată: permi. ca două sau mai multe procese să fol. același seg. de mem. Crearea sau deschiderea accesului la mp. se face cu opel sistem shmget; int shmget (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 27) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 28) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 29) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 30) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 31) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 32) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 33) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 34) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 35) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 36) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 37) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 38) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 39) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 40) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 41) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 42) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 43) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 44) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 45) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 46) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 47) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 48) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 49) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 50) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 51) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 52) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 53) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 54) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 55) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 56) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 57) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 58) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 59) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 60) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 61) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 62) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 63) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 64) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 65) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 66) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 67) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 68) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 69) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 70) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 71) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 72) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 73) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 74) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 75) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 76) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 77) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 78) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 79) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 80) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 81) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 82) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 83) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 84) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 85) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 86) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 87) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 88) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 89) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 90) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 91) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 92) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 93) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 94) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 95) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 96) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 97) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 98) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 99) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm. 100) IPC semafoare: a fast întreg pt a aloc. f. de intrare și deschiderea se face prin opel sistem msgget(f), funcția întoarce un întreg par. f., descriptorul care, num. msgid. Repunerea unui mesaj se face prin opel shmput (key-t chei, int lungime, int flag), rel. un întreg. descriptor al segm. unit. shm.