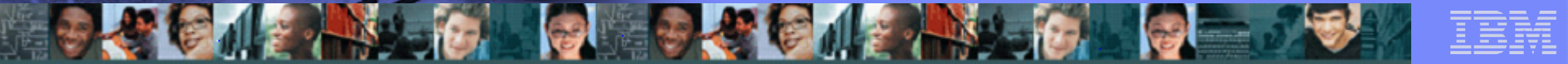


IBM Academic Initiative



Modulo 3 : Il sistema operativo z/OS (parte prima)

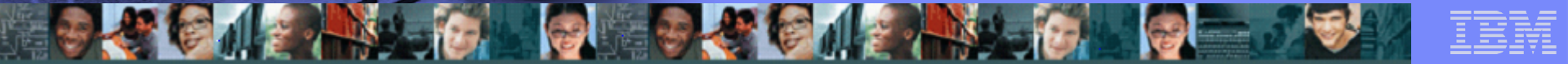




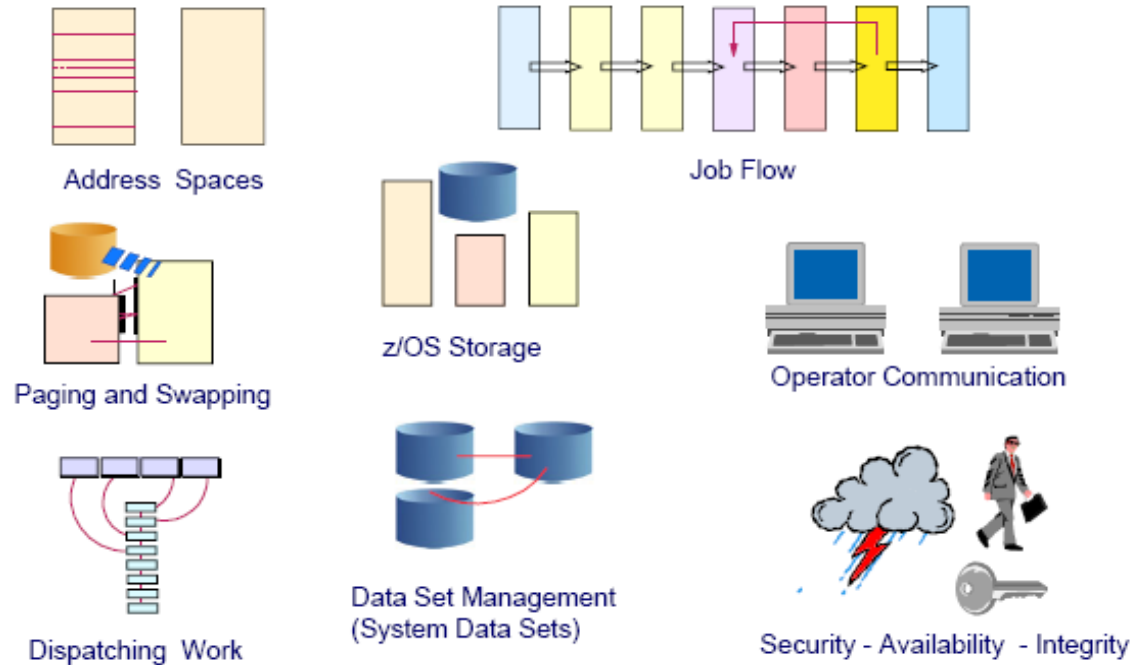
Cosa e' z/OS ?

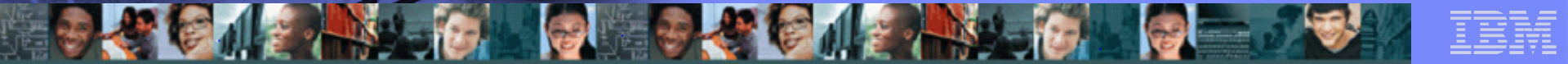
- Il Sistema Operativo **piu' usato sui Mainframes**
 - **trattamento di grandi moli di dati**
 - **Progettato per Sistema Operativo a 64 bit**
 - **Ideale per il trattamento di grandi volumi di lavoro**
 - **Progettato per servire molti utenti concorrenti**
 - **Progettato per Servire applicazioni 'critiche' in maniera Sicura.**

- Struttura Interna di z/OS
 - **Comprende moduli, macro e Componenti di Sistema.**
 - **Usa la struttura PSW.**
 - **Usa Tecniche di multi programmazione e processori multipli**
 - **Le informazioni riguardanti il Sistema , le risorse , e le applicazioni attive sono contenute in 'Blocchi di Controllo'**
 - **Gestione della Memoria suddivisa in:**
 - **Reale** (memoria centrale)
 - **Ausiliaria** (usualmente su disco)



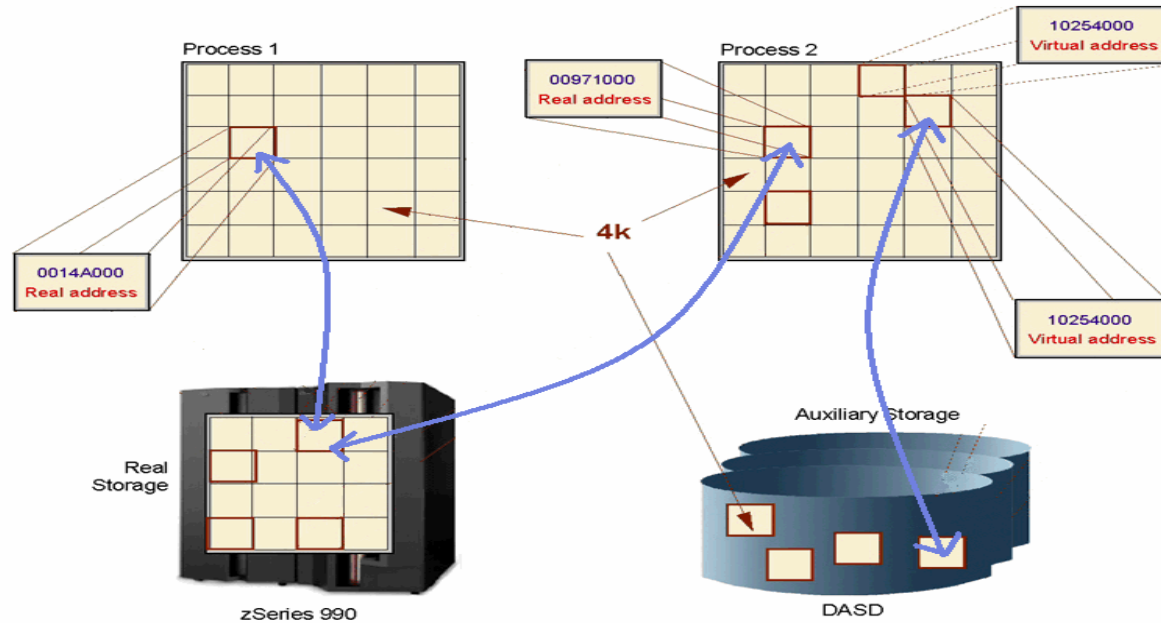
Le componenti importanti del S.O z/OS

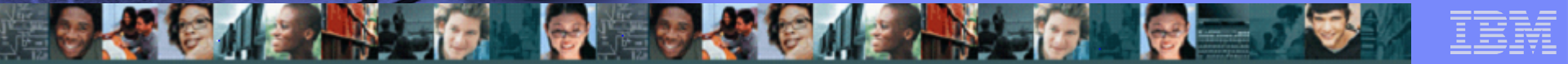




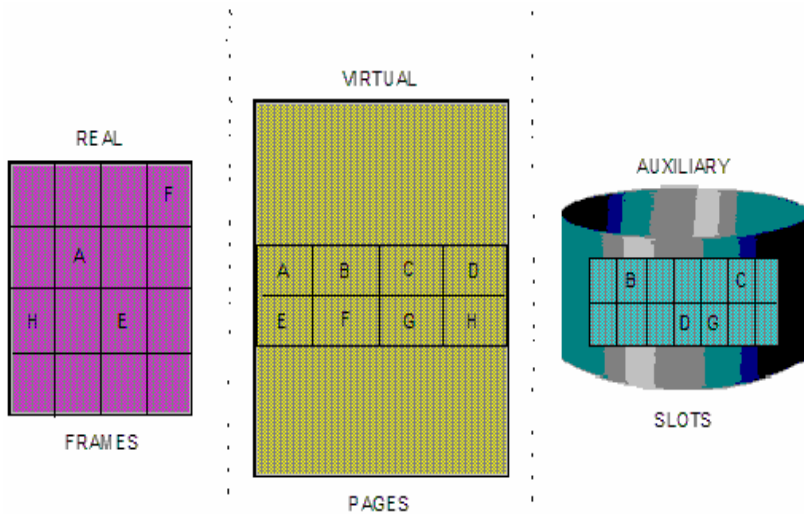
Come lavora la memoria Virtuale

La memoria Virtuale e' divisa in 'pagine ' da 4 KB.





Pages, Frames, and Slots – Stealing - Swapping



Il Programmatore Applicativo non percepisce la paginazione quanto egli 'vede' la memoria come **contigua**.

Il Sistema z/OS tende a tenere in Memoria Reale un numero adeguato di pagine.

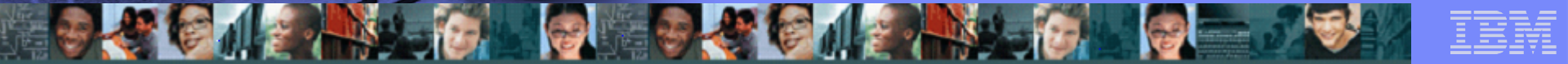
Per raggiungere questo obiettivo alcune pagine vengono periodicamente paginate fuori (page-out) in modo da tenere delle frames libere per potere paginare dentro (page-in) le nuove pagine richieste.

Tale processo prende il nome di 'Page Stealing'

Un particolare processo di paginazione prende il di 'Swapping':

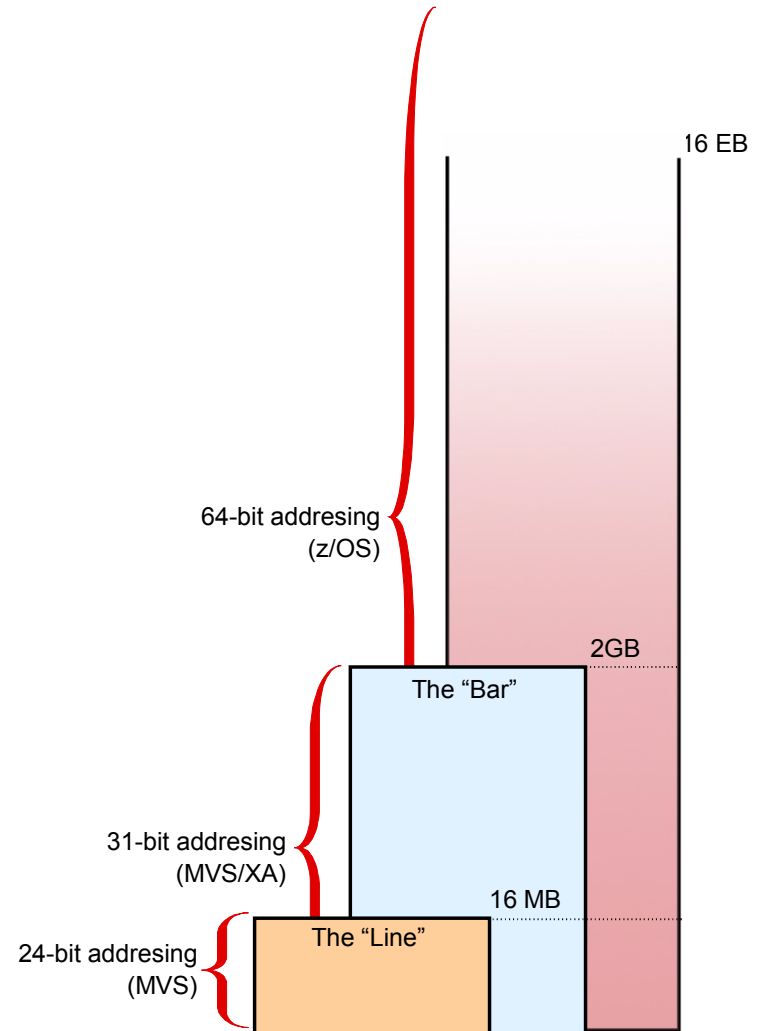
Esso si ha quando il Sistema pone fuori dalla Memoria Centrale tutte le pagine che costituiscono un Address Space, e poi le ricarica tutte quando servono di nuovo.

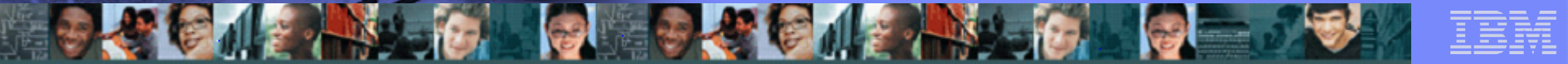
Lo Swapping e' tipico di programmi che hanno lunghi periodi di attesa (long wait)



IL Concetto di Memoria Virtuale – Address Space

- La memoria Virtuale e' un'artificio che consente ad ogni utente di eseguire i programmi avendo disposizione tutta la memoria definita nell'architettura.
- Tutta la Memoria Virtuale che e' utilizzata da un programma viene raggruppata in una struttura Logica denominata 'Address Space'.
- Ogni programma utente o di Sistema ,attivo e' rappresentato da un Address Space.
- Gli Address Spaces possono contenere programmi utente o programmi di Sistema in questo caso avremo:
 - User Address Spaces
 - System Address Spaces
- La 'dimensione' di un Address Space e' determinata dal numero di bit usati per l'indirizzamento 'ADDRESS MODE'. z/OS supporta Address Spaces a 24, 31 o 64bit.



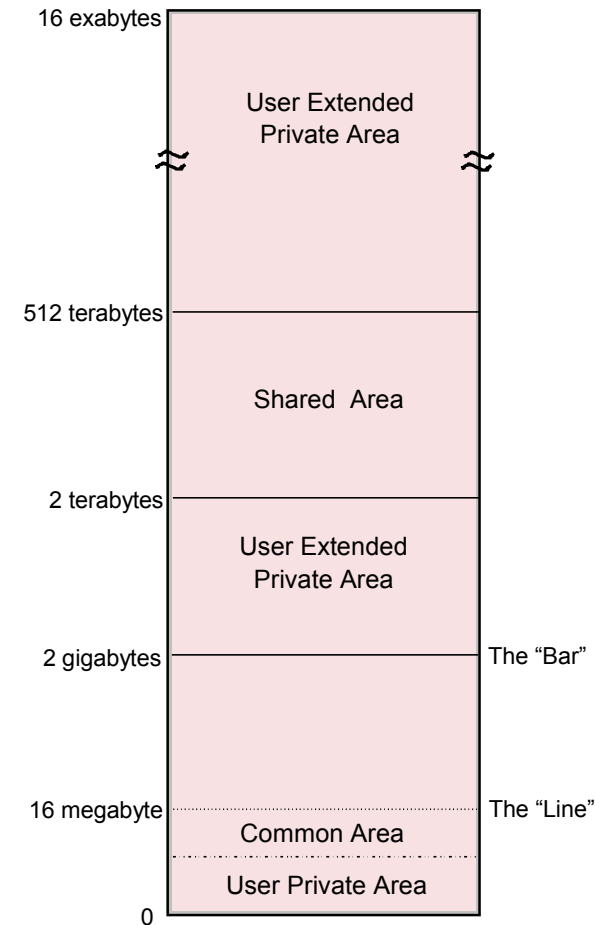


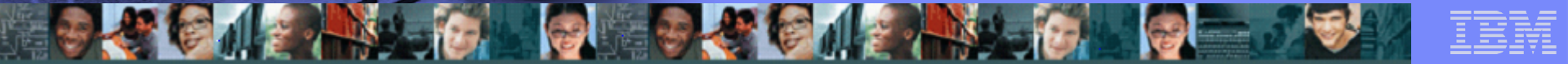
IL Contenuto di un Address Space (Address Space MAP)

- z/OS fornisce ad ogni 'utente' un suo Address Space e fa in modo che esso **sia isolato dagli altri** Address Spaces e che all'interno di esso i dati siano distinti dalle istruzioni.
- Poiché un Address Space rispecchia tutto il contenuto di un Programma esso includerà anche parti di Sistema Operativo, dati, ed elementi di servizio.

Address Spaces gestiti da z/OS:

- System Address Spaces : Vengono creati all' avvio del Sistema ed hanno funzioni di servizio e di gestione per tutti gli altri Address Spaces creati successivamente.
- SubSystem Address Spaces: Sono creati da alcuni programmi detti sottosistemi che servono a gestire Basi di Dati, o prodotti middleware come Monitor, Gestori di Transazioni etc.
- TSO/E Address Spaces creati per ogni nuovo utente che si collega in modo interattivo al Sistema z/OS
- User Address Spaces: creati per ogni programma (batch job).





Sommario delle Capacita' di Base di z/OS

L'uso degli Address Space:

assicura l'integrita' di ogni dato presente a fronte di una moltitudine di utenti.

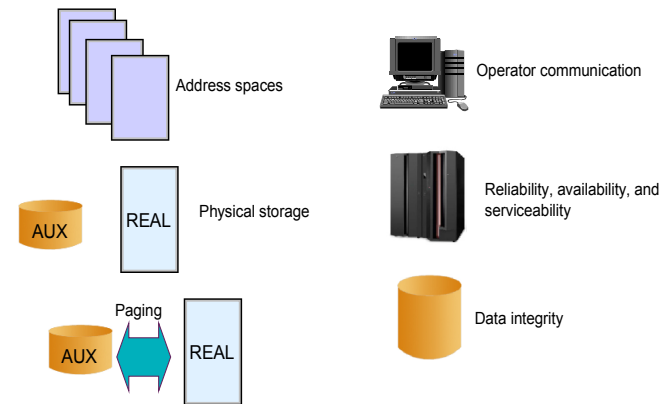
Consente l'elaborazione di un grande numero di lavori di tipo 'batch' con capacita' di bilanciamento automatico.

consente l'uso di un dispositivo di Sicurezza integrato a livello di :

- **Accesso**
- **Uso delle Risorse**
- **Uso di Programmi**

multiprogramming
multiprocessing
Interrupt processing
Dispatching works
Serializzazione dell'uso di risorse

z/OS



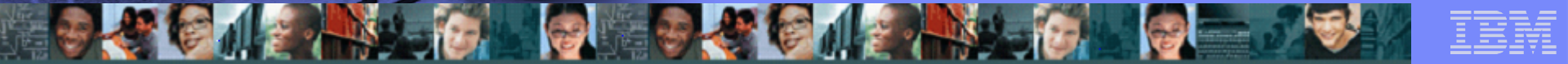
•Consente di attivare **molte programmi di comunicazione** allo stesso tempo.

•Fornisce funzioni avanzate per la **ripartenza a fronte di interruzioni accidentali.**

•Consente la gestione contemporanea di tipi diversi di lavori bilanciandone l'esecuzione.

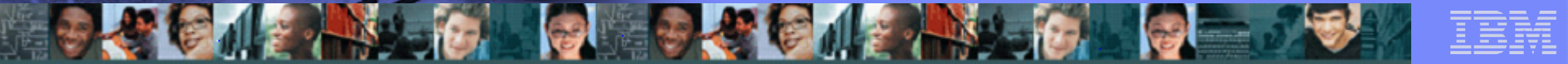
•Consente di gestire una **grande quantita' di dispositivi di I/O** (Dischi Nastri Stampanti Terminali)

•Può essere **controllato manualmente o in maniera automatica** da appositi programmi.



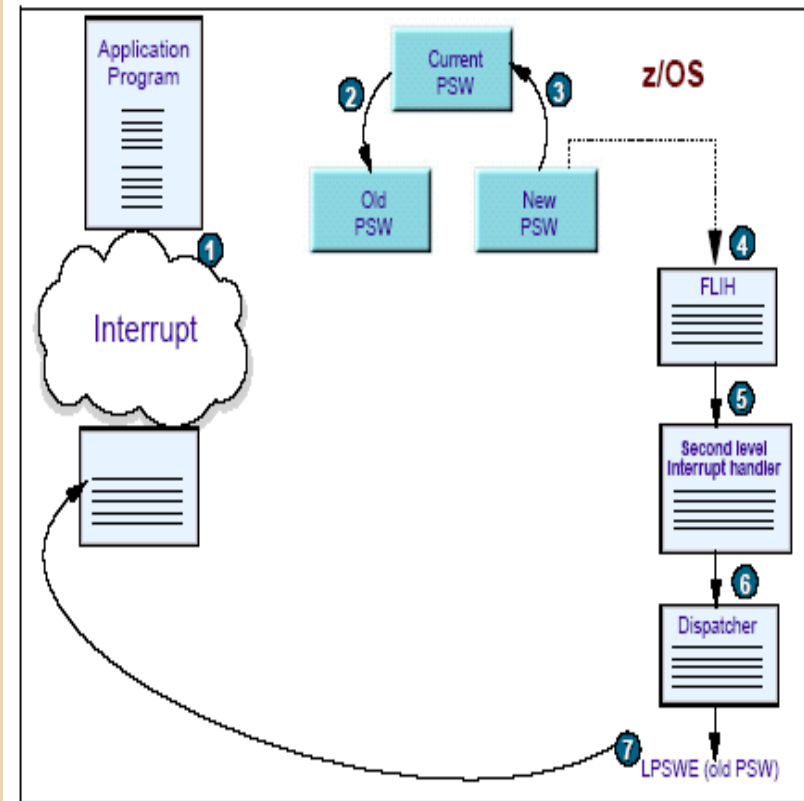
z/OS -Supervisione dell'esecuzione di lavoro nel sistema

- Processo di Interruzione (interrupts)
- Creazione di unità di lavoro eseguibili (dispatchable UoW)
- Processo di Dispatching
- Serializzazione delle risorse riusabili

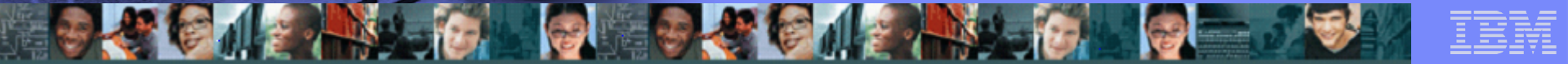


Elementi di z/architecture – Interruzioni (interrupts)

1. Mentre il programma utente sta eseguendo e' viene interrotto da un segnale di interruzione
2. salvando la PSW corrente nella locazione di memoria ad essa dedicata(*)
3. L'HW carica la New PSW il cui indirizzo punta al FLIH (first Level Interrupt handler)
4. Lo z/OS passa il controllo al FLIH che analizza il tipo di Interrupt e individua l'Handler opportuno SLHI (second level interrupt handler)
5. Viene dato il controllo allo SLHI che decide cosa fare a seconda del tipo di interrupt occorso
6. IL controllo viene poi passato al Dispatcher che ricarica il vecchio contesto (LPSWE old PSW)



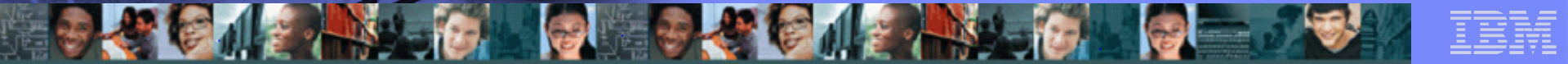
* I registri non sono salvati. Tale azione e' delegata al software



z/OS – Scheduler e Dispatcher

Il processo di scheduling/dispatching di un address space e altre task in z/OS si occupano

2. Reperire le unità di lavoro
3. Eseguire le unità di lavoro
4. Salvare lo stato per le unità di lavoro



z/OS – Gestione dei processi

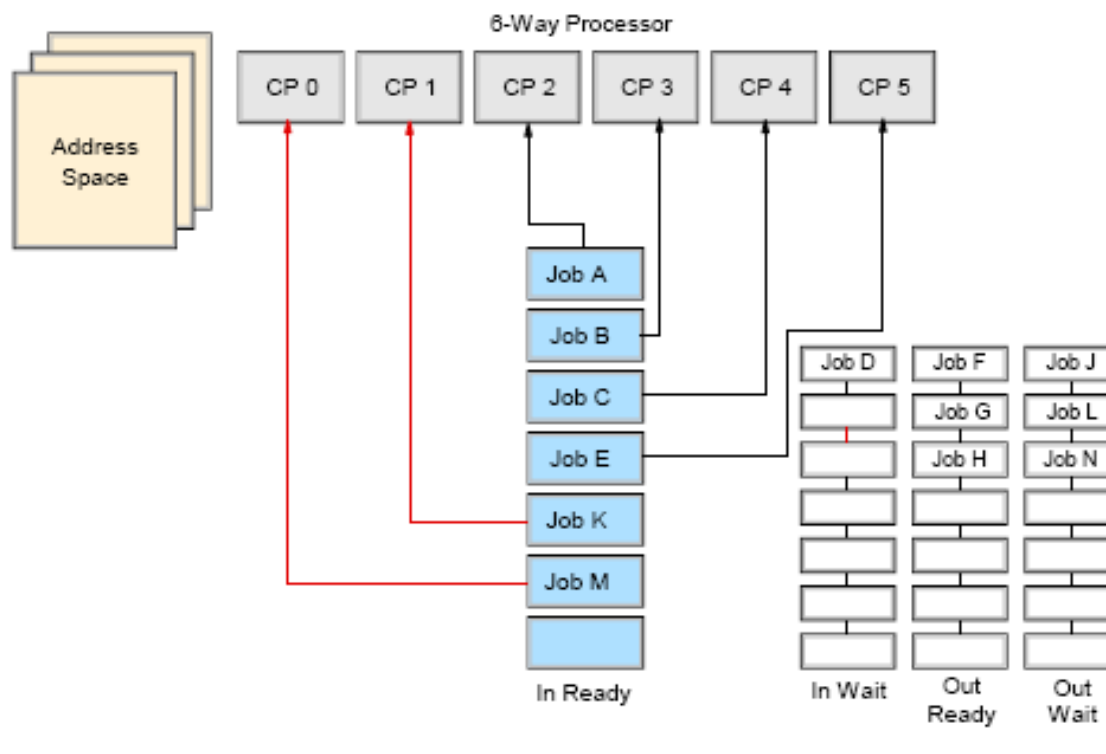
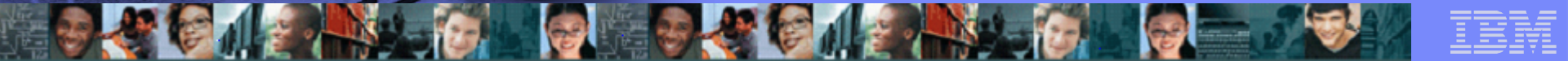
Le unità oggetto di dispatching all'interno di z/OS sono

- Task Control Block (TCB)

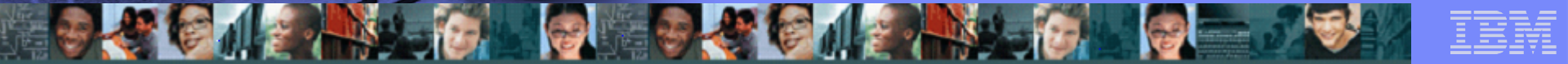
- Unità di lavoro che eseguono in un address space – Programmi utente
- Un address space può avere più TCB ad esso associati.
- L'unità di lavoro TCB è creata quando un programma esegue una macro denominata ATTACH

- Service Request Block (SRB)

- Usualmente iniziata dal supervisore (quindi in Supervisor State) per mezzo di codice iniziato in un address space per effettuare un azione che interessa un altro Address Space (es I/O operation)
- L'unità di lavoro SRB e' non interrompibile (non preemptive) con qualche eccezione
- L'unita' di lavoro e' ad alta priorita'

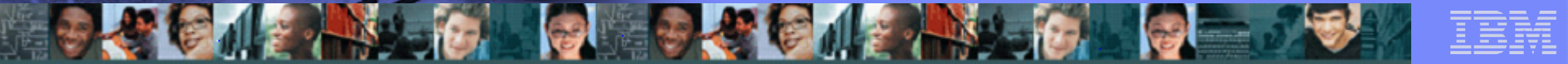


- ▶ An address space can be in any one of four queues:
 - IN-READY - In real storage and waiting to be dispatched
 - IN-WAIT - In real storage but waiting for some event to complete
 - OUT-READY - Ready to execute but swapped out
 - OUT-WAIT - Swapped out and waiting for some event to complete
- ▶ Only IN-READY work will be selected for dispatching.



z/Os - Processo di dispatching

- Code di Dispatching- catene di pagine di controllo
- Ogni ASCB (Blocco di Controllo che identifica l'Address Space) contiene la sua catena di SRB e TCB
- Mentre avviene il Dispatching la old PSW e' salvata in Prefixed Save Area e viene caricata la New PSW nella Current PSW .
- Interrupts – che innescano lo SWAP di PSW sono analizzati dal FLIH .
- Quando il controllo è restituito al Dispatcher esso seleziona la unità di lavoro con maggiore priorità con questo ordine:
 1. Hardware recovery
 2. SRB con priorità globale
 3. ASCB
 4. SRB con priorità locale
 5. TCB

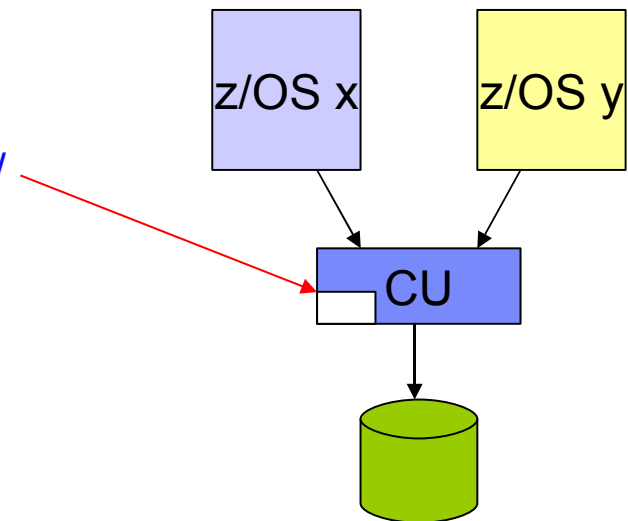


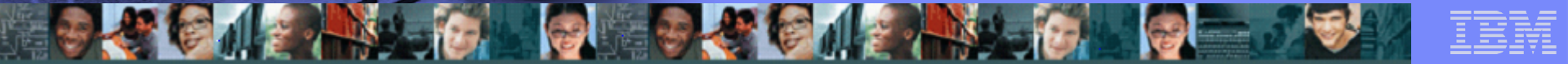
z/OS Processo di Serializzazione di risorse riusabili

In un ambiente di multiprogrammazione quasi ogni sequenza di istruzioni può essere .

Se quell'insieme di istruzioni manipola o modifica una risorsa (p.e. un control block o un file) il sistema operativo deve prevenire uso di quella risorsa da parte di altri programmi fino a che il programma interrotto non completi la sua esecuzione su quella risorsa

2. Serializzazione tra sistemi disaccoppiati RESERVE/RELEASE
3. Enqueueing ENQ/DEQ (Problem state)
4. Locking LOCK (Supervisor state)

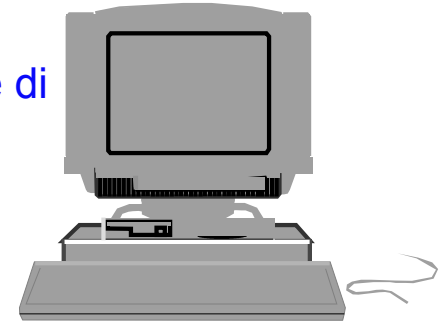


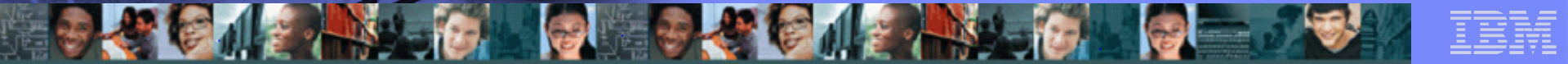


Come si puo' interagire con z/OS

Esistono tre interfacce Utente possibili mediante le quali da un Terminale Video Connesso ad un Sistema z/OS e' possibile interafire col Sistema:

- TSO
 - Consente agli utenti di accreditarsi al Sistema e di dare una serie di comandi diretti (Nativi)
- ISPF
 - Consente di accedere al Sistema mediante una interfaccia a caratteri o grafica basata su liste di possibili scelte e percorsi guidati. Fornisce anche programmi per l'immissione di dati, testi o altri programmi (EDITORS). Puo' essere attivata da Terminali a caratteri/pagine (3270) o da Personal Computers
- z/OS UNIX shell and utilities
 - Fornisce una Shell di tipo UNIX (Korn Shell) per accedere al Sistema.

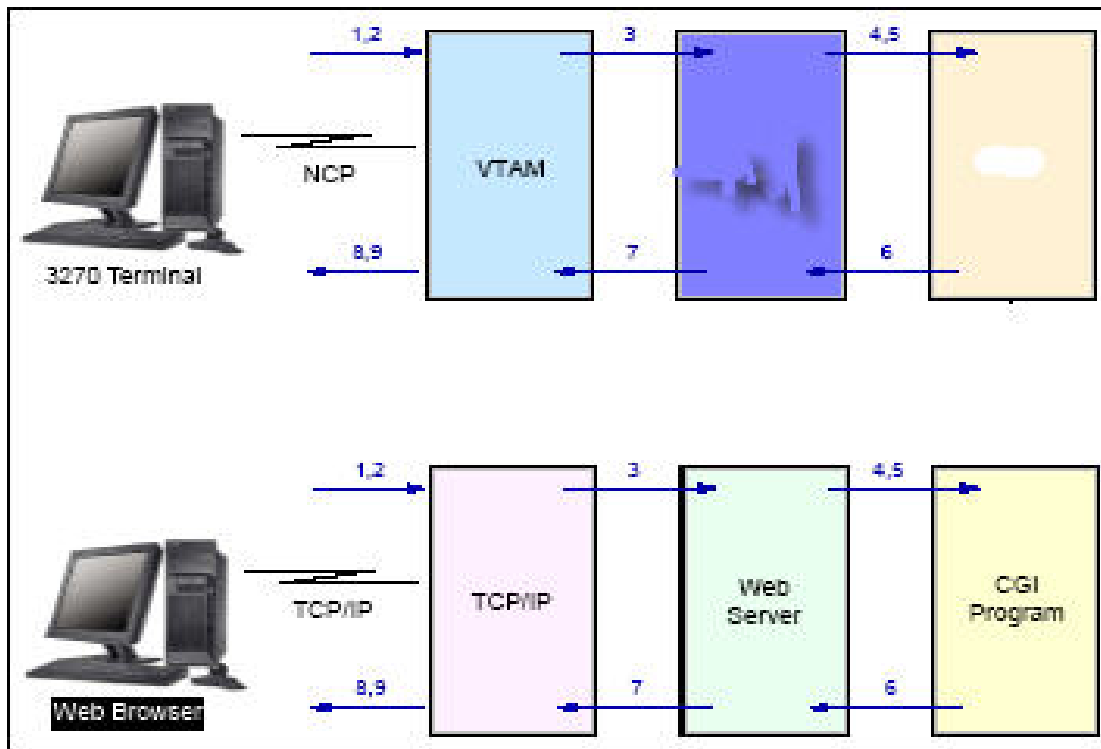


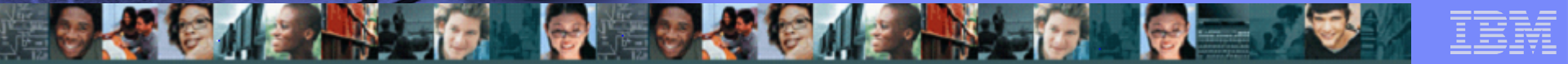


Interfacce utente con applicazioni in ambito z/OS

Esistono varie interfacce utente per accedere ad applicazioni in ambito z/OS.

- Terminali 3270 **Terminali a carattere/ pagine**
- Terminali ASCII **Terminali a carattere/ pagine**
- Browser **emulazione grafica su Personal Computer**





TSO overview

TSO/E (Time Sharing Option/Extension) e'uno strumento di multiprogramming.

Permette di creare sessioni interattive con z/OS a partire da un terminale video.

Fornisce la capacita' ad ogni utente di disporre di un proprio accesso e di un ambiente di esecuzione isolato.

Ospita l'ISPF (Interactive System Productivity Facility) . Programma a menu che opera in ambiente TSO/E

Fornisce un accesso al Sistema tramite un utenza dotata di identificativo e password

Richiede un Address Space per ogni utente che accede al Sistema.

In fase di Accesso al TSO/e (logon) fasi di accesso tramite opportuni parametri caratterizza la sessione di lavoro attraverso il terminale IBM3270 (o la sessione emulata TN3270).

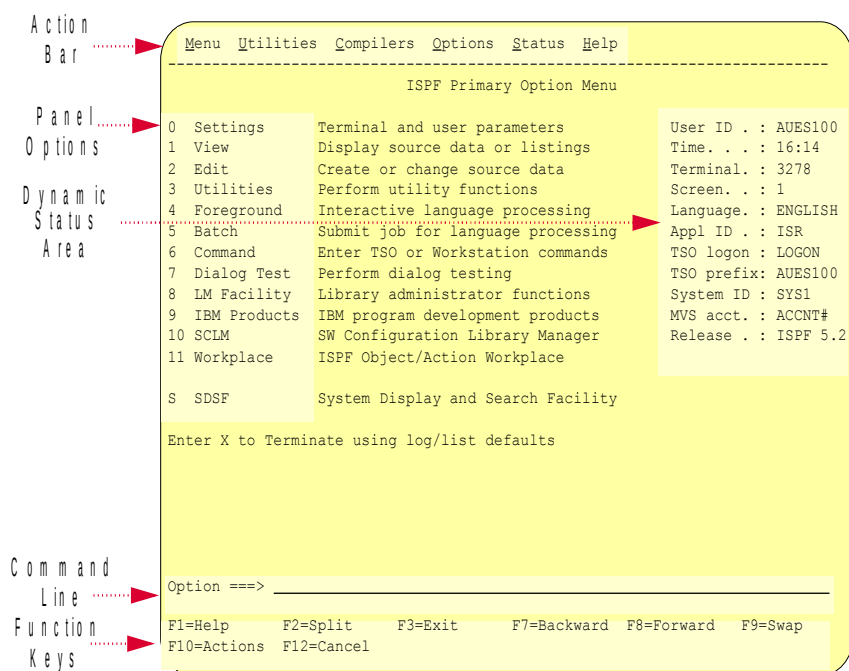
```
-----TSO/E LOGON-----
.
.
.
.  Enter LOGON parameters below:      RACF LOGON parameters:
.
.  Userid  ==> ZPROF
.
.  Password ==>
.                                     New Password ==>
.
.  Procedure ==> IKJACCN1
.                                     Group Ident ==>
.
.  Acct Nbr ==> ACCNT#
.
.  Size    ==> 880000
.
.  Perform ==>
.
.  Command ==>
.
.  Enter an 'S' before each option desired below:
.      -Nomail   -Nonotice  -Reconnect  -OIDcard
.
.  PF1/PF13 ==> Help  PF3/PF15 ==> Logoff  PA1 ==> Attention  PA2 ==> Reshow
.  You may request specific help information by entering a '?' in any entry field
```

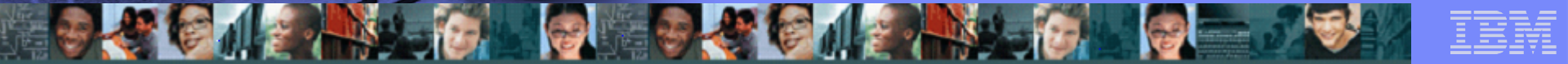
```
READY
time
TIME-11:42:11 AM. CPU-00:00:00 SERVICE-788 SESSION-00:25:25 JULY 21,2003
READY
```

Le caratteristiche di accesso degli utenti TSO/E possono essere personalizzate dal Programmatore del Sistema o in parte direttamente dall'utente tramite Profili individuali o di gruppo.

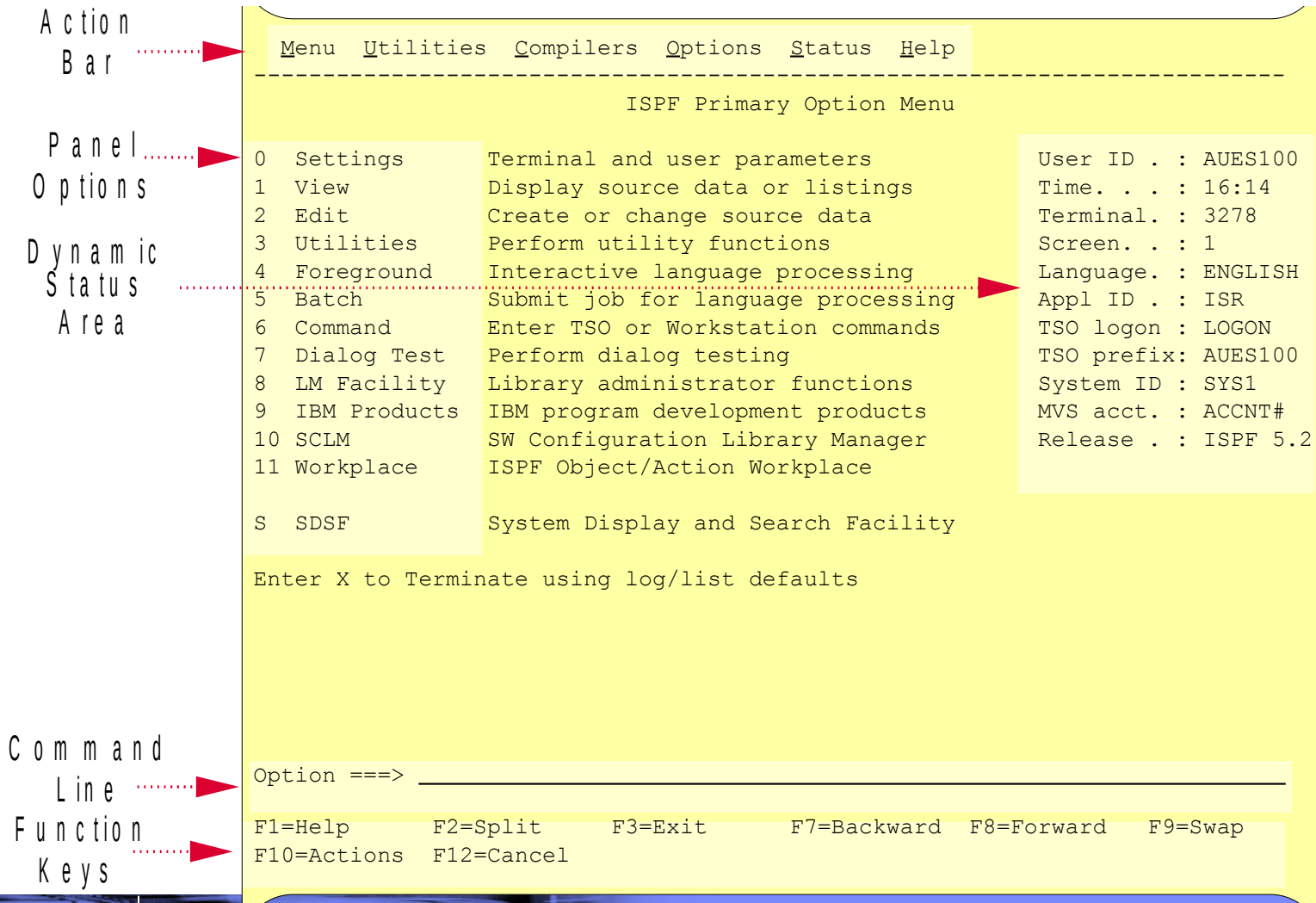
ISPF overview

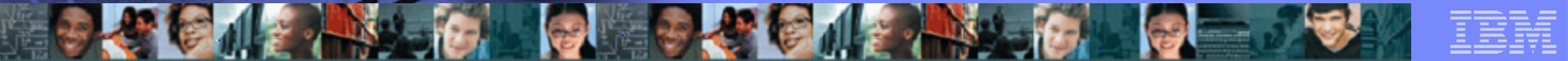
- ISPF ovvero Interactive System Productivity Facility
- ISPF e' una interfaccia basata su menu per consentire un accesso piu' semplice verso z/OS .
- L'ambiente ISPF viene eseguito attraverso il TSO .
- ISPF fornisce molte funzioni
 - programmi di utilita'
 - Editors per ' di modificare programmi e dati mediante accesso diretto ai files
 - consente di dare comandi al sistema.
 - Consente eseguire allocazioni di Archivi, richiamo di programmi ed esecuzione di lavori.
 - Consente di interfacciare UNIX files systems
- Il tutto viene gestito attraverso I meccanismi di Sicurezza integrati nel Sistema.





General structure of ISPF panels





z/OS UNIX interactive interfaces

- L'interfaccia di tipo UNIX (USS) e' stata introdotta su z/OS a meta' degli anni 90 (Standard POSIX)
- La Unix Shell puo' essere usata per:
 - Richiamare Shell Scripts and Utilities.
 - Scrivere programmi di utilita' tipici di UNIX con appropriati linguaggi.
 - Eseguire 'shell scripts' e programmi scritti in linguaggio C.

La Shell UNIX puo' essere richiamata mediante:

- Terminale IBM 3270 o un suo emulatore .
- Terminale connesso via TCP/IP usando comandi :
 - RLOGIN
 - TELNET
- DA TSO con i 'bidge commands' :
 - OMVS
 - ISHELL

```
# Improve shell performance
if [ -z "$STEPLIB" ] && tty -s;
then
    export STEPLIB=none
    exec sh -L
fi

# Set the time zone as appropriate.
TZ=EST5EDT

# Set a default command path, including current working dir (CDW)
PATH=/bin:.

# Sets the path environment variables
LIBPATH=/lib:/usr/lib:.
MANPATH=/usr/man/%L
NLSPATH=/usr/lib/nls/msg/%L/%N

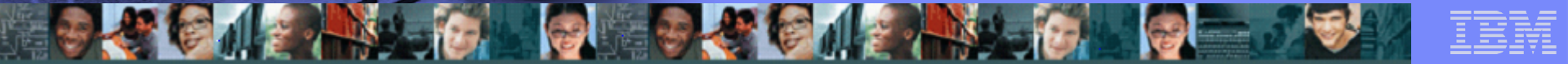
# Sets the language
LANG=C

# Sets the name of the system mail file and enables mail notification.
MAIL=/usr/mail/$LOGNAME

# Export the values so the system will have access to them.
export TZ PATH LIBPATH MANPATH NLSPATH MAIL LANG

# Set the default file creation mask - umask
umask 022

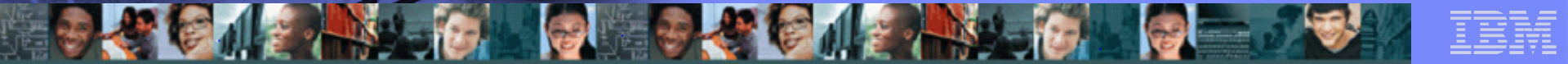
# Set the LOGNAME variable readonly so it is not accidentally modified.
readonly LOGNAME
```



Cosa si intende per 'batch processing'?

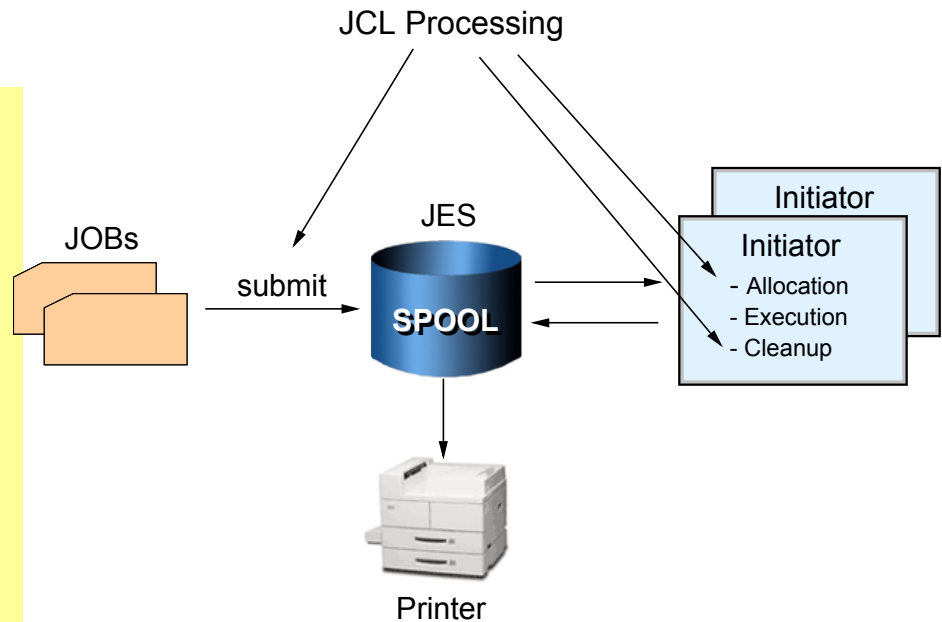
PROCESSO A BLOCCHI (Batch Processing)

- **Per 'Batch Processing' si intende una tipologia di lavoro costituito da un programma o un insieme ordinato e controllato di programmi, che possono essere eseguiti:**
 - **Con una interazione umana minima o nulla.**
 - **In un certo momento o al ricorrere di una certa circostanza o in particolari periodi della giornata o dell'anno in modo estemporaneo o schedato.**
- **Un lavoro BATCH (Batch Job)' esegue una certa quantita' di elaborazioni per produrre in uscita:**
 - **Gli stessi dati modificati o trasformati**
 - **Altri dati di sintesi o di dettaglio sullo stesso o su altri archivi**
 - **Stampe**
 - **logs**
- **Una volta richiesta al Sistema l'esecuzione di un Job Batch usualmente il Sistema non richiede altre interazioni umane fino al suo completamento.**
- **L' utente puo' verificare la conclusione del Job esaminandone I risultati o controllando su un apposito archivio prodotto, detto Job Log I vari passi compiuti ed I loro effetti.**
- **Un Job Batch puo' essere sospeso e riavviato , puo' concludersi positivamente o puo' essere interrotto per Errori (ABbnormalEND o ABEND)**



IL Job Entry SubSystem (JES)

- Nel Sistema Operativo z/OS il JES fornisce il supporto ai lavori Batch gestendone anche i risultati in apposite 'code'.
- JES gestisce i seguenti aspetti del processo:
 - Riceve le richieste di lavori Batch .
 - Ne stabilisce le modalità di esecuzione (scheduling), controllando e risolvendo eventuali conflitti.
 - Controlla i risultati e le emissioni (OUTPUT).

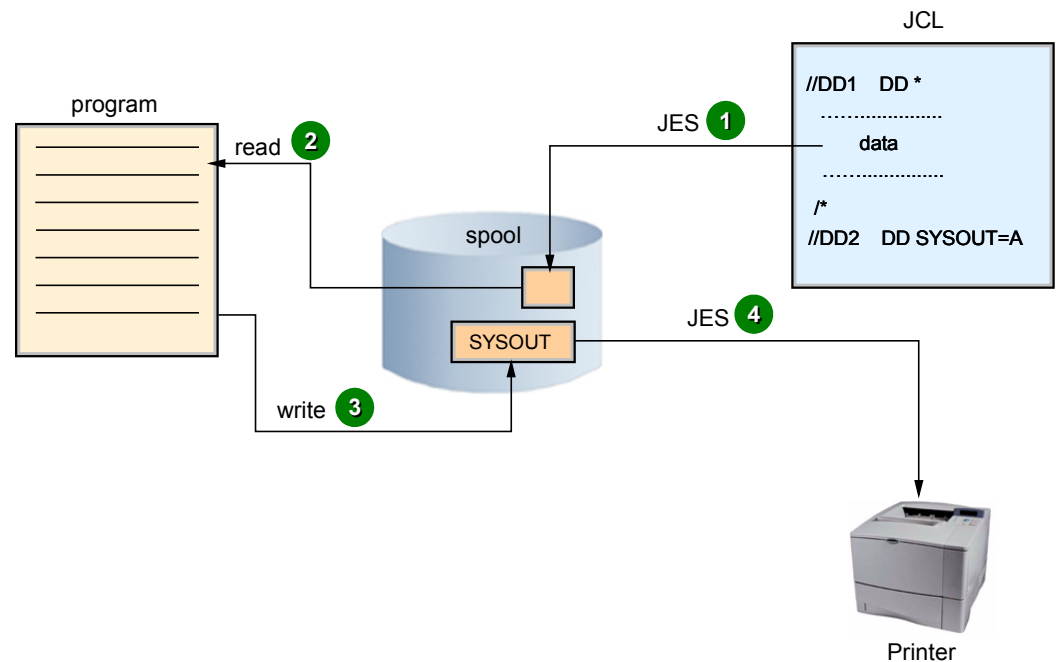


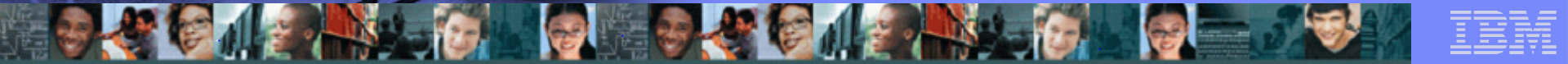
- Si definisce 'Spooling' la capacità di un sistema operativo di accodare (queuing) le richieste di esecuzione (input queue) ed i risultati dei Job (in genere stampe) in modo da essere consultabili e poi stampate solo se necessario (Output Queue)
- JES2 usa degli appositi spazi su disco per eseguire tali funzioni.
- Input jobs and printed output per diversi JOB sono immagazzinati in un unico contenitore detto 'spool data set'.

Spooling and Job Control Language

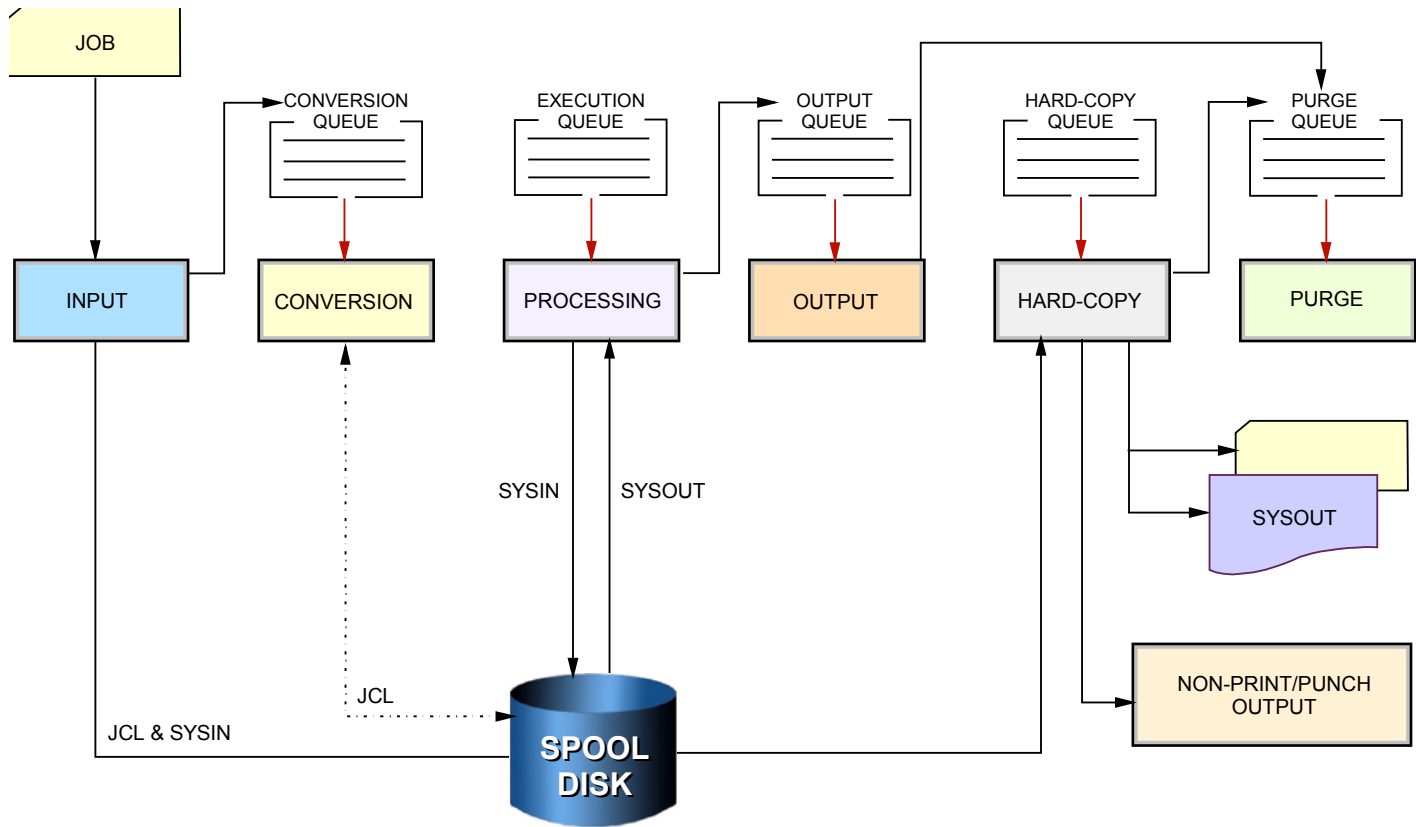
Si definisce **JCL** (Job Control Language)
il linguaggio usato per :

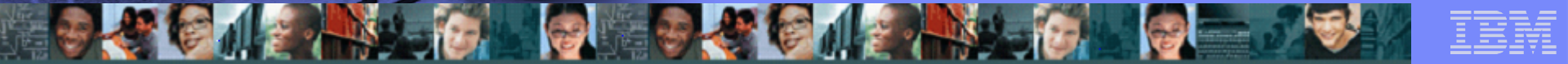
- Richiedere al Sistema le **Risorse** necessarie ad una lavoro BATCH
- Indicare i **programmi** da eseguire
- Indicare i **parametri** da utilizzare nei programmi
- Definire le caratteristiche di **sicurezza**
- Definire **caratteristiche di esecuzione**





Job flow del JES attraverso il Sistema





Workload Manager (Goal Mode)

- Workload Manager (WLM) e' una componente del Sistema Operativo z/OS.

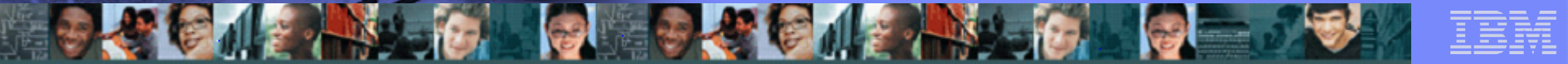
Esso consente l'esecuzione **simultanea di programmi concorrenti** in accordo con politiche **complessive di utilizzo** ed in modo tale che venga raggiunto un obiettivo complessivo di **prestazione (Goal)**.

Politica

Service Class	Per	Goal	Value	Imp
CICS	1	Avg RT	0.1 s	2
IMS	1	Avg RT	10 s	3
TSO	1	Avg RT	0.1 s	3
	2	Avg RT	1.0 s	3
	3	Avg RT	3.0 s	4
Batch	1	ExVel	10	5

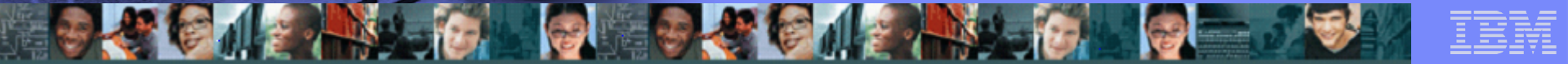
- il Workload Manager e' in grado di:
 - **agire sulle risorse del Sistema (Come CPU e Memoria)**
 - **modificare l'ordine di esecuzione dei Job, la priorit  dei programmi in esecuzione**
 - **modificare i meccanismi di uso della memoria Virtuale , della Paginazione**
 - **modificare la priorit  nell'uso dei dispositivi di I/O.**

Quando un Sistema z/OS opera sotto il controllo del WLM si dice che esso sta operando in Goal Mode



Obiettivi del Capitolo B.3

- 2. z/OS Overview
- 4. User Interfaces
- 6. Job Entry Subsystem
- 8. Job Control Language / Spool Display and Search Facility
- 10. Data Management on z/OS
- 12. Data Bases on z/OS
- 14. Design Applications for z/OS
- 16. Programming Languages for z/OS
- 18. Transactional Systems



System Display and Search Facility (SDSF)

■ **SDSF e' una componente del Sistema Operativo z/OS in grado di :**

- **Mostrare gli utenti Collegati**
- **Immettere Comandi di Sistema**
- **Hold, release, cancel, and purge jobs**
- **Eseguire il Monitoring di JOB durante il processo**
- **Visualizzare le stampe prima di produrle**
- **Controllare l'ordine di immissione dei lavori (Controllare l'ordine di emissione dei lavori (output Queue)**
- **Controllare i dispositivi collegati al Sistema**
- **Mostrare il LOG di Sistema**

```
Display Filter View Print Options Help
-----
ISFPCU41 ----- SDSF PRIMARY OPTION MENU -----
COMMAND INPUT ==> _ SCROLL ==> PAGE

DA  Active users          INIT  Initiators
I   Input queue          PR   Printers
O   Output queue         PUN  Punches
H   Held output queue    RDR  Readers
ST  Status of jobs       LINE Lines
                               NODE Nodes
                               SO   Spool offload
                               SP   Spool volumes

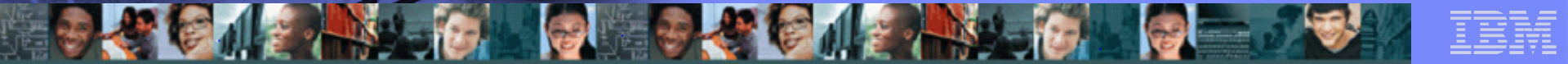
LOG  System log          ULOG  User session log
SR   System requests
MAS  Members in the MAS
JC   Job classes
SE   Scheduling environments
RES  WLM resources
ENC  Enclaves
PS   Processes

END  Exit SDSF

Licensed Materials - Property of IBM

5694-A01 (C) Copyright IBM Corp. 1981, 2002. All rights reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=IFIND      F6=B00K
F7=UP        F8=DOWN       F9=SWAP       F10=LEFT      F11=RIGHT     F12=RETRIEVE
```



Un Esempio : La visualizzazione delle Stampe

Screen 1

```
Display Filter View Print Options Help
-----
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 44 LINE 1-1 (1)
COMMAND INPUT ==> SCROLL ==> PAGE
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
NP JOBNAME JobID Owner Prty C ODisp Dest Tot-Rec Tot-
?_ MIRIAM2 JOB26044 MIRIAM 144 T HOLD LOCAL 44
```

Screen 2

```
Display Filter View Print Options Help
-----
SDSF JOB DATA SET DISPLAY - JOB MIRIAM2 (JOB26044) LINE 1-3 (3)
COMMAND INPUT ==> SCROLL ==> PAGE
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
NP DDNAME StepName ProcStep DSID Owner C Dest Rec-Cnt Page
JESMSG LG JES2 2 MIRIAM T LOCAL 20
JESJCL JES2 3 MIRIAM T LOCAL 12
JESYSMSG JES2 4 MIRIAM T LOCAL 12
```

SDSF: Display active users (DA)

```

Display Filter View Print Options Help
-----
SDSF DA SC67 SC67 PAG 0 SIO 7 CPU 6/ 7 LINE 1-25 (64)
COMMAND INPUT ===> SCROLL ===> PAG
PREFIX=* DEST=LOCAL OWNER=* SORT=JOBNAME/A
NP  JOBNAME  STEPNAME  PROCSTEP  JOBID  OWNER  C  POS  DP  REAL  PAGING  SIO
*MASTER*                STC06373 +MASTER+  NS  FF  1369  0.00  0.00
ALLOCAS  ALLOCAS                NS  FF  190  0.00  0.00
ANTAS000 ANTAS000 IEFPROC        NS  FE  1216  0.00  0.00
ANTMAIN  ANTMAIN IEFPROC        NS  FE  4541  0.00  0.00
APPC     APPC    APPC          NS  FE  2653  0.00  0.00
ASCH     ASCH    ASCH          NS  FE  267  0.00  0.00
BPXOINIT BPXOINIT BPXOINIT        LO  FF  315  0.00  0.00
CATALOG  CATALOG  IEFPROC        NS  FF  1246  0.00  0.00
CICSPAAY CICSPAAY CICS520  STC06504 STC  NS  FE  4330  0.00  0.00
CONSOLE  CONSOLE                NS  FF  597  0.00  0.00
DFRMM    DFRMM    IEFPROC        STC06363 STC  NS  FE  510  0.00  0.00
DFSMSHSM HSMSC67  DFSMSHSM  STC13178 STC  NS  FE  6199  0.00  0.00
DUMPSRV  DUMPSRV  DUMPSRV        NS  FF  160  0.00  0.00
FTPDMVS1 STEP1                STC06477 STC  LO  FF  470  0.00  0.00
FTPDOE1  STEP1                STC06475 FTPDOE  LO  FF  469  0.00  0.00
GRS      GRS          NS  FF  894  0.00  0.00
IEFSCHAS IEFSCHAS        NS  FF  25  0.00  0.00
IMWEBSUF IMWEBSUF WEBSRV  STC15245 WEBSRV  IN  FE  15T  0.00  0.00

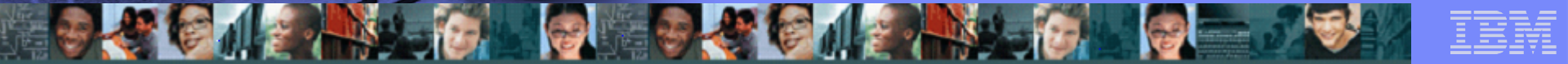
```

SDSF: Status panel

```

Display Filter View Print Options Help
-----
SDSF STATUS DISPLAY ALL CLASSES                                LINE 1-24 (3281)
COMMAND INPUT ===>                                           SCROLL ===> PAGE
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
NP  JOBNAME  JobID   Owner   Prty Queue   C  Pos  SAff  ASys  Status
BARTR1DB JOB06472 BARTR1   10 EXECUTION A           HOLD
BARTR1DB JOB06479 BARTR1   10 EXECUTION A           HOLD
BARTR1DB JOB06561 BARTR1   10 EXECUTION A           HOLD
BARTR1DB JOB06565 BARTR1   10 EXECUTION A           HOLD
BARTR1DB JOB06568 BARTR1   10 EXECUTION A           HOLD
BARTR1DB JOB06588 BARTR1   10 EXECUTION A           HOLD
BARTTEP1 JOB09138 BART    10 EXECUTION A           HOLD
TWSSTD3  TSU26002 TWSSTD3  15 EXECUTION           SC64 SC64
KMT1     TSU26024 KMT1    15 EXECUTION           SC64 SC64
MIRIAM   TSU26043 MIRIAM   15 EXECUTION           SC64 SC64
HAIMO    TSU26050 HAIMO    15 EXECUTION           SC63 SC63
BARTR4   TSU26051 BARTR4   15 EXECUTION           SC63 SC63
RAVI     TSU26052 RAVI     15 EXECUTION           SC63 SC63
BARTR2   TSU26060 BARTR2   15 EXECUTION           SC63 SC63
VBUDI    TSU26062 VBUDI    15 EXECUTION           SC64 SC64
SYSLOG   STC24863 +MASTER+ 15 EXECUTION           SC63 SC63
RACF     STC24871 RACF     15 EXECUTION           SC63 SC63
SYSLOG   STC24931 +MASTER+ 15 EXECUTION           SC64 SC64
RACF     STC24941 RACF     15 EXECUTION           SC64 SC64
OPTSO    STC24857 STC      15 EXECUTION           SC63 SC63
OAM      STC24858 STC      15 EXECUTION           SC63 SC63
RMF      STC24855 STC      15 EXECUTION           SC63 SC63
SDSF     STC24862 STC      15 EXECUTION           SC63 SC63 ARMELEM
ASCHINT  STC24867 STC      15 EXECUTION           SC63 SC63

F1=HELP  F2=SPLIT  F3=END    F4=RETURN  F5=IFIND  F6=BOOK
F7=UP    F8=DOWN   F9=SWAP   F10=LEFT   F11=RIGHT F12=RETRIEVE
  
```

Utilities & System Libraries

- **z/OS include anche altri programmi di servizio detti utilities.**
- **Utilities forniscono molte semplici ed utili funzioni**
- **Un Elenco dei programmi di utilita' e' fornito nella Appendice C del Testo.**
- **Spesso gli Utenti scrivono delle loro particolari utilities o le acquistano presso venditori di Software Indipendenti (ISV)**
- **z/OS e' dotato di un cospicuo numero di librerie di Sistema (System Libraries) che contengono i programmi, le provcedure i jobs e i parametri per il normale funzionamento del sistema operativo**

Alcuni esempi di utilities:

IEBGGENER Copia un Dataset sequenziale

IEBCOPY Copia un Dataset partizionato

IDCAMS Lavora con datasets VSAM

Alcuni esempi di librerie standard dette 'system libraries':

SYS1.PROCLIB contiene le procedure JCL distribuite con lo z/OS

SYS1.PARMLIB contiene i parametri di controllo dello z/OS e di alcuni prodotti installati

SYS1.LINKLIB contiene molti dei moduli eseguibili del Sistema

SYS1.LPALIB contiene moduli eseguibili di sistema caricati alla inizializzazione dello z/OS. Tali moduli forniscono servizi per gli utenti.