z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM Software Group

TPF Users Group Fall 2005

Z/TPF Enhancements and Structure Changes of Interest for Applications Programmers

Name : Rick Matela
Venue : Applications Development Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Agenda

- Program Packaging
- Application Stack Structure
- Program Linkage

Program Packaging

Definitions

- All z/TPF ECB Programs will be loaded as Shared Objects
- All C DLLs, DLMs, and LLMs will be known as CSOs, C Shared Objects.
- All BAL programs will be referred to as BSOs, BAL Shared Objects.

BSO Structure

- Initially, 1 BSO = 1 assembler program
- Eventually, 1 BSO = many assembler programs
  - Refer to BSO program CVAA
    CVAA is comprised of:
    CVAA
    CVAJ
    CVAH
    CVAI
    CVAD
    CVAO
- An assembler program may reside in more than one BSO.

BSO Structure

- 4.1 program structure location 0 is the start of the listing and the program.

BEGIN

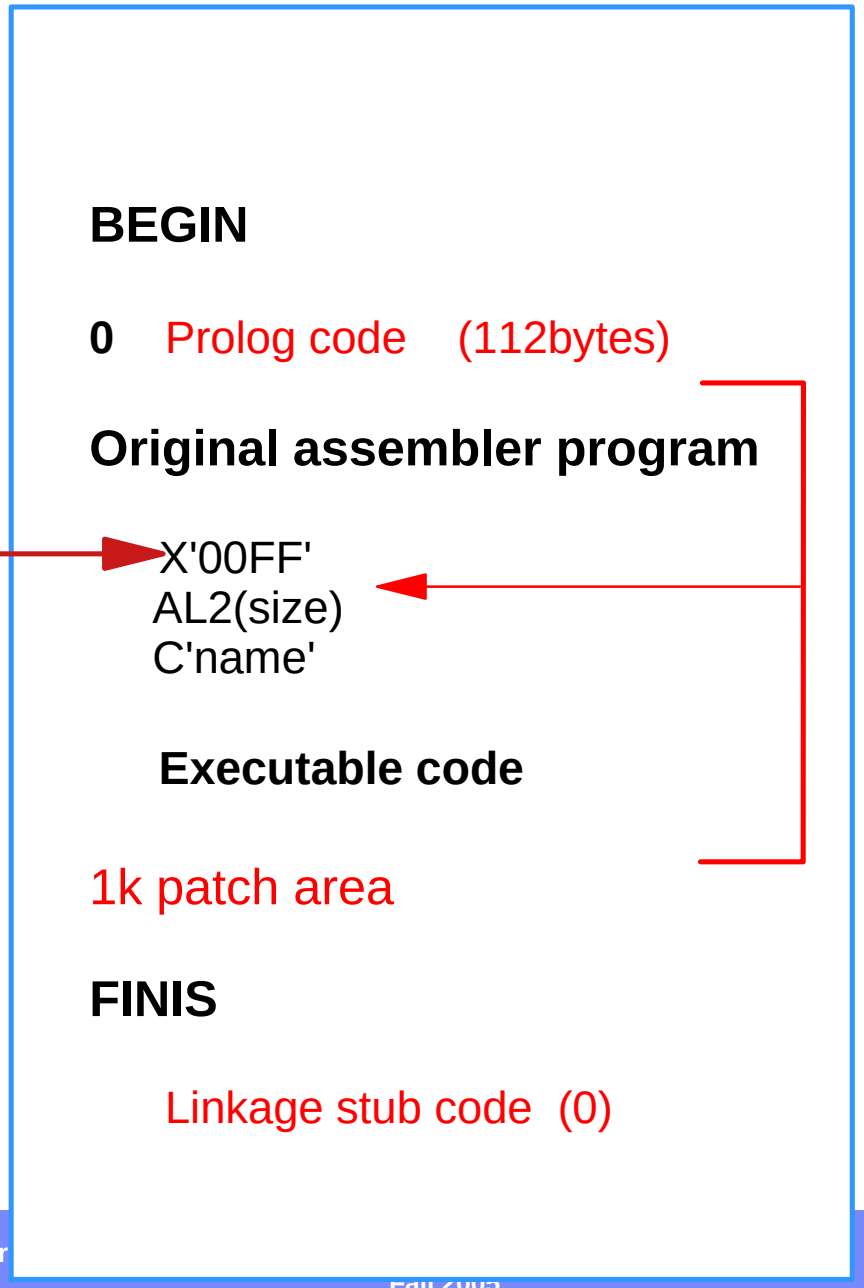
0 X'00FF'
AL2(size)
C'name'

Executable code

FINIS

BSO Structure

- z/TPF program structure
  - BEGIN macro adds prolog code
  - R8 still points to original program object
  - A 1K patch area is added to the end of the program
  - The FINIS macro generates all kinds of stub code



BSO Support

- Benefits
  - Faster linkage between programs (internal)
  - >4K programs supported
  - Application stack provided
  - Register usage expanded (R0-R8, R10-R13)
- Considerations
  - All programs use the allocation characteristics defined for the BSO

New Support

- BEGIN -
  - VERSION= no longer necessary
  - BASE= defines base register, 1 or multiple or none
  - AMODE= defines mode of operation 31or 64bit
  - TV= defines transfer vectors
  - TYPE= defines type of program
    - Executable - normal E-Type program
    - Data - program used as data record
    - Keypoint - system keypoint record

New Support

- BEGIN -
  - BASELESS= macros can generate baseless code
  - EREGSAVE= macros will save R10-R13 (DEFBC)
  - TRNAME= defines setting of trace/tagging field
  - DSECT=, USEREG= defines user area on stack
- APSTKC - used with DSECT=

Example

- Getting Application stack space through BEGIN:

```
BEGIN NAME=CVZZ,IBM=YES,
      DSECT=WORKAREA,USEREG=R4

SAVE_IT  DS    F
LAST_ADR DS    F

APSTKC
```

New Support

- FINIS - No parameters but.... generated code
  - ENTxC/BACKC linkage code
  - ALASC, FLIPC, CALLC, and GLOBC macro code
- ENTxC macros - branch to linkage code in FINIS
- BACKC - branch to linkage code in FINIS
- ALLOC - allocates space on the stack

```
ALLOC     SIZE=1000,ADDR=R2,ERROR=NO_ROOM
```

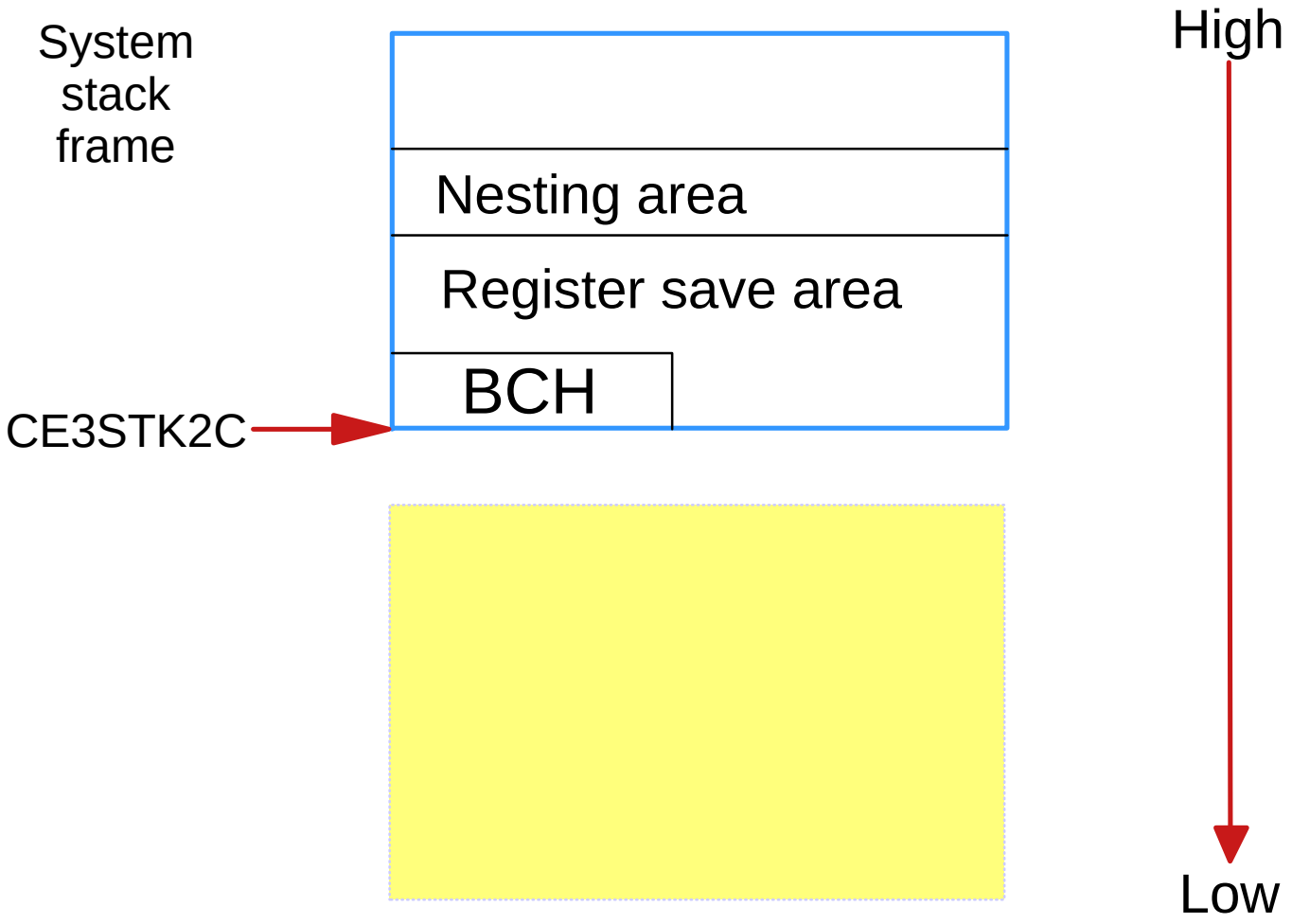
New Support

- CPROC and CALLC - call 'C' functions from assembler code

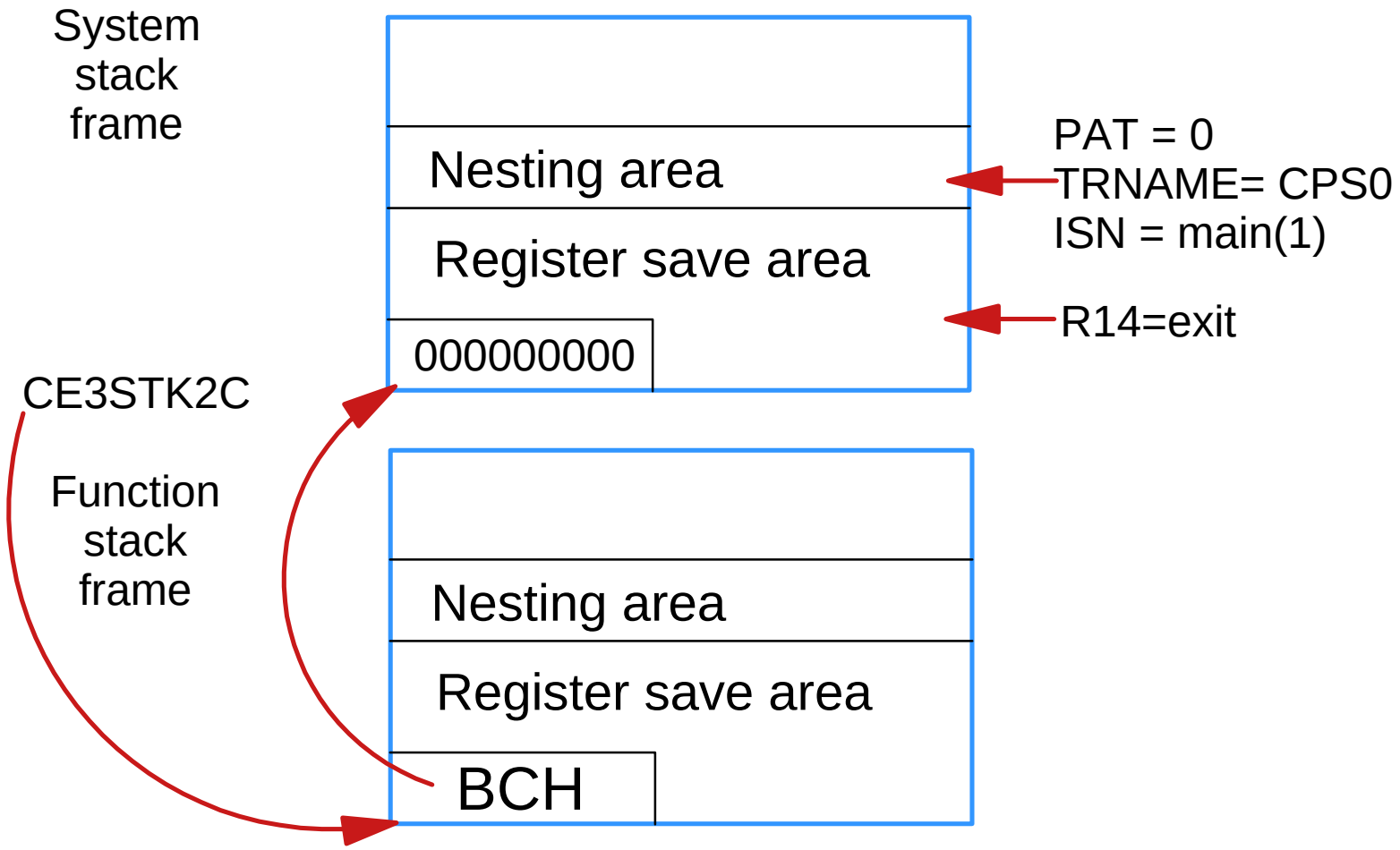
```
CPROC RETURN=p,getcc,(i,i,i)  Call C function
LGHI  R2,D1
XGR   R3,R3
LGHI  R4,L1
CALLC getcc(R2,R3,R4)

CPROC RETURN=i,CVZ5,(p,i,i,i) Call C program
LA    R14,KEYPT               Record Type for CVZ5
LA    R7,DE                   Read Level for CVZ5
LA    R0,KPTR_NO_HOLD         No Hold Flag for CVZ5
LA    R1,DF                   Validate KPTR Flag for CVZ5
LLGFR R14,R14                 Clean up ptr field for call
CALLC CVZ5(R14,R7,R0,R1)      Read KPtr Rec
```

Application Stack Structure

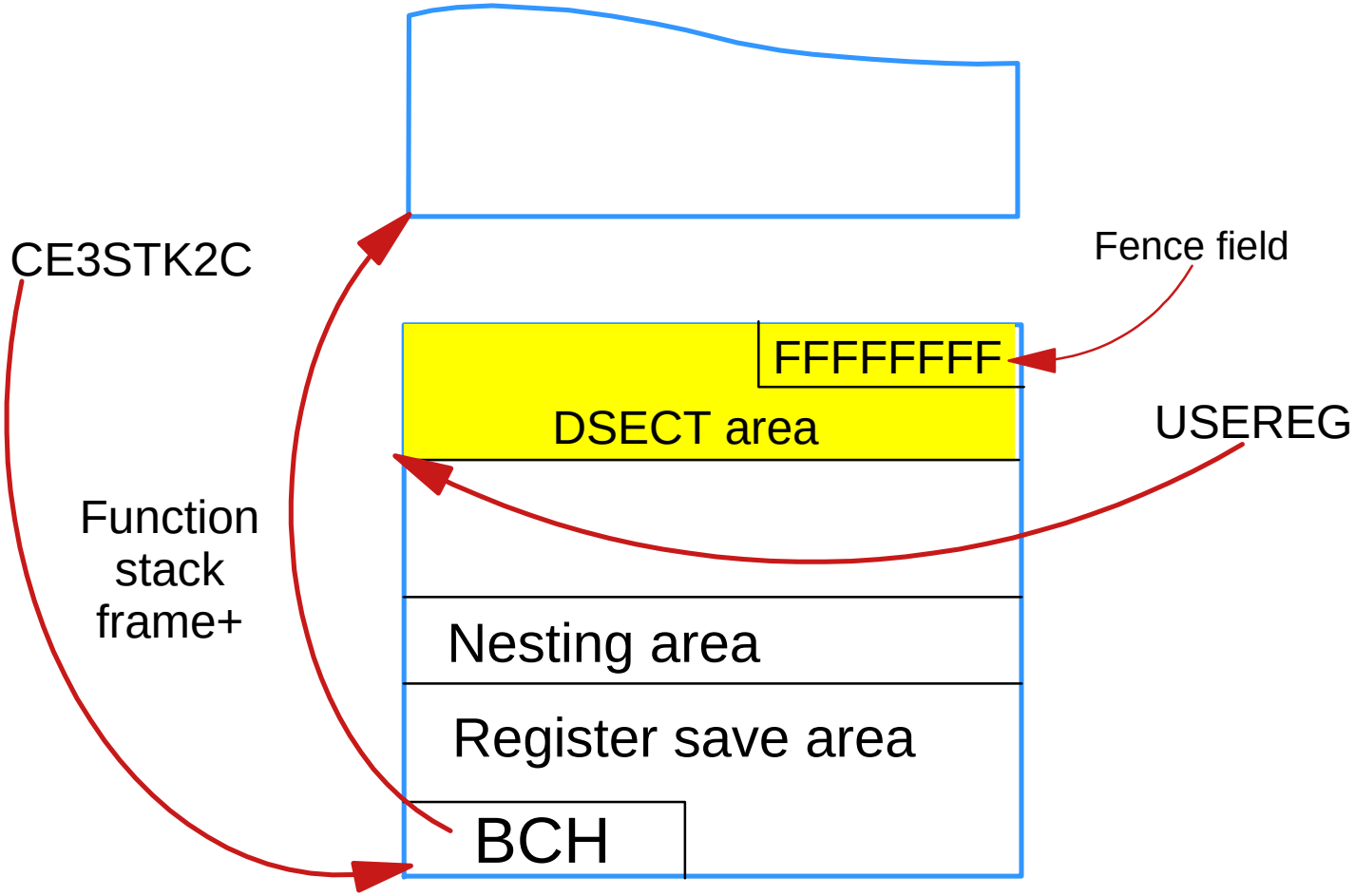
Application Stack Structure



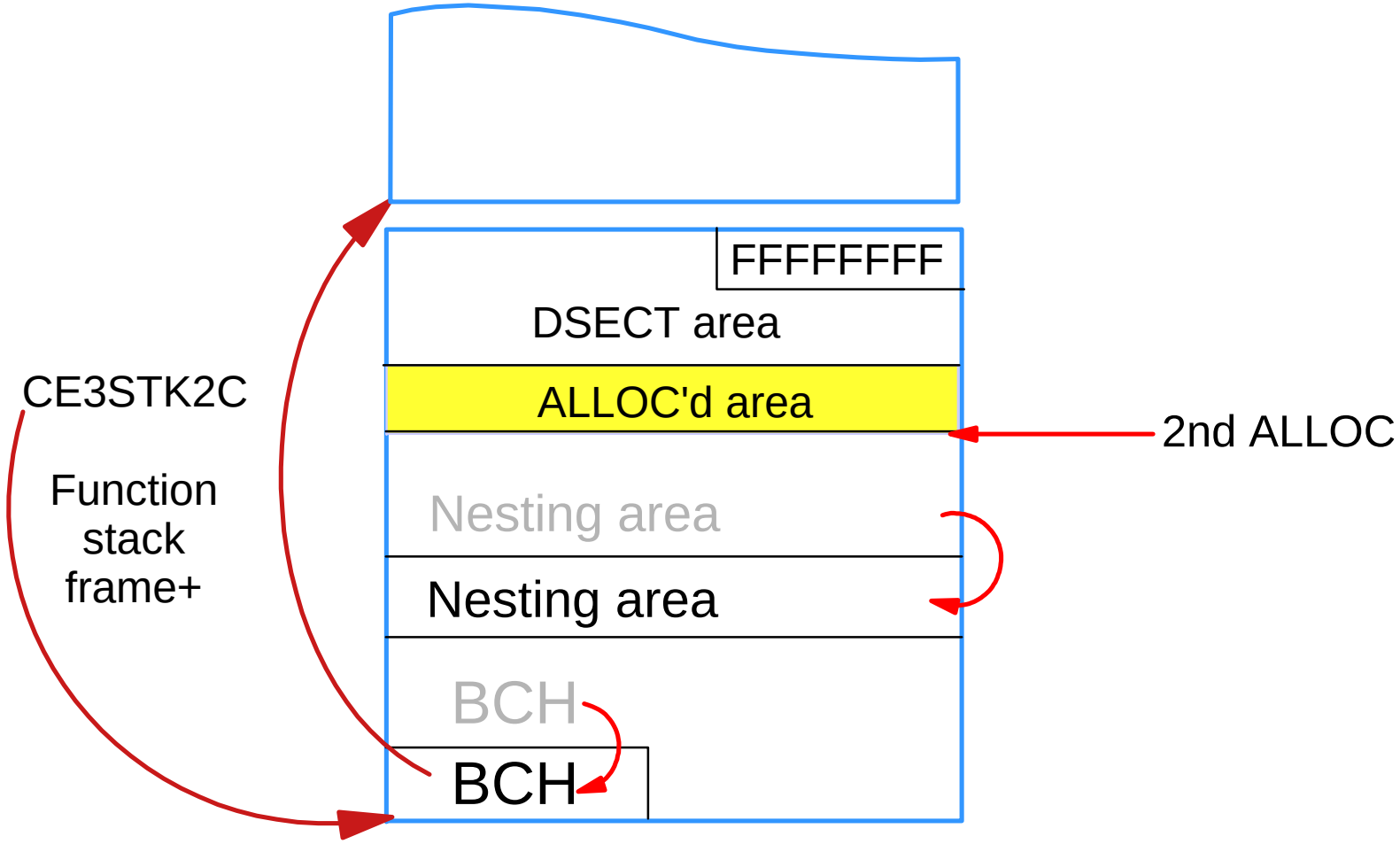
Application Stack Structure



Example
Getting Application stack space through BEGIN:



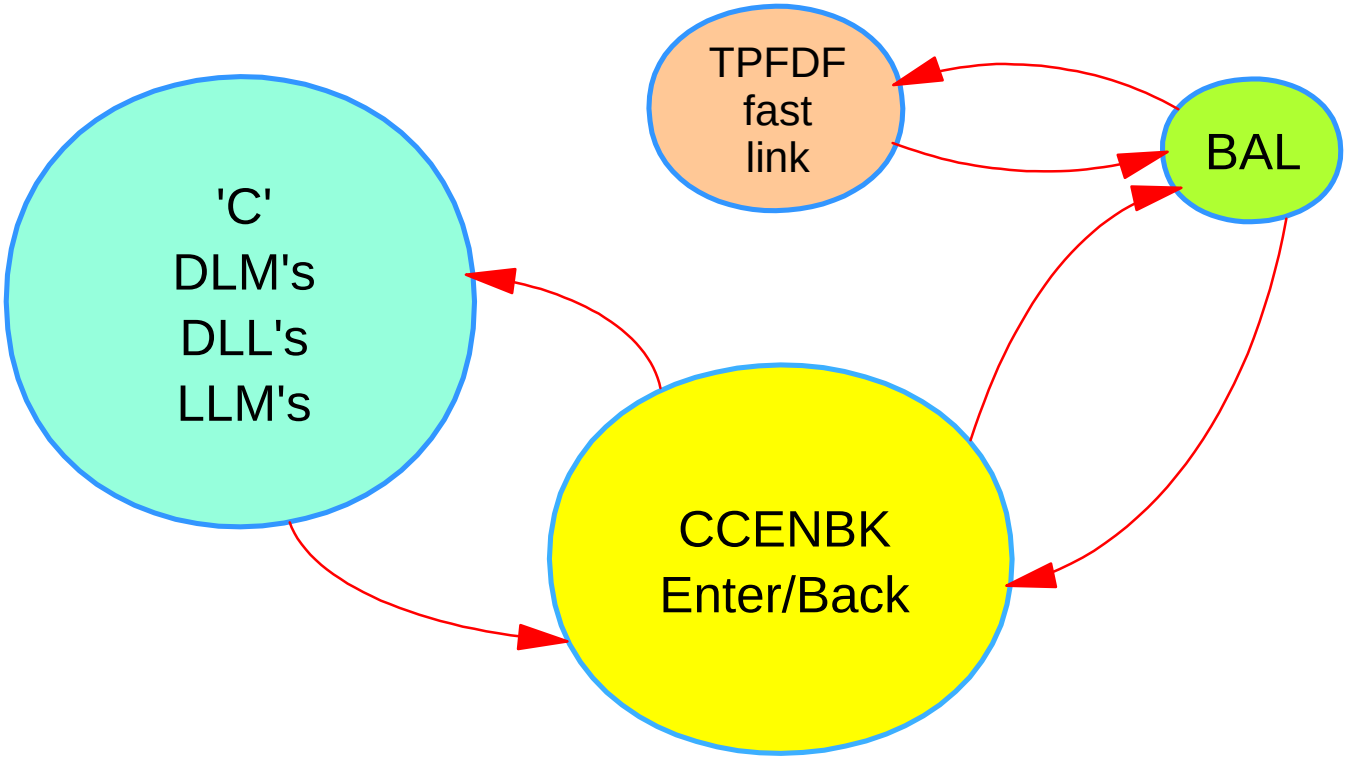
Example
Getting Application stack space through ALLOC:



Program Linkage

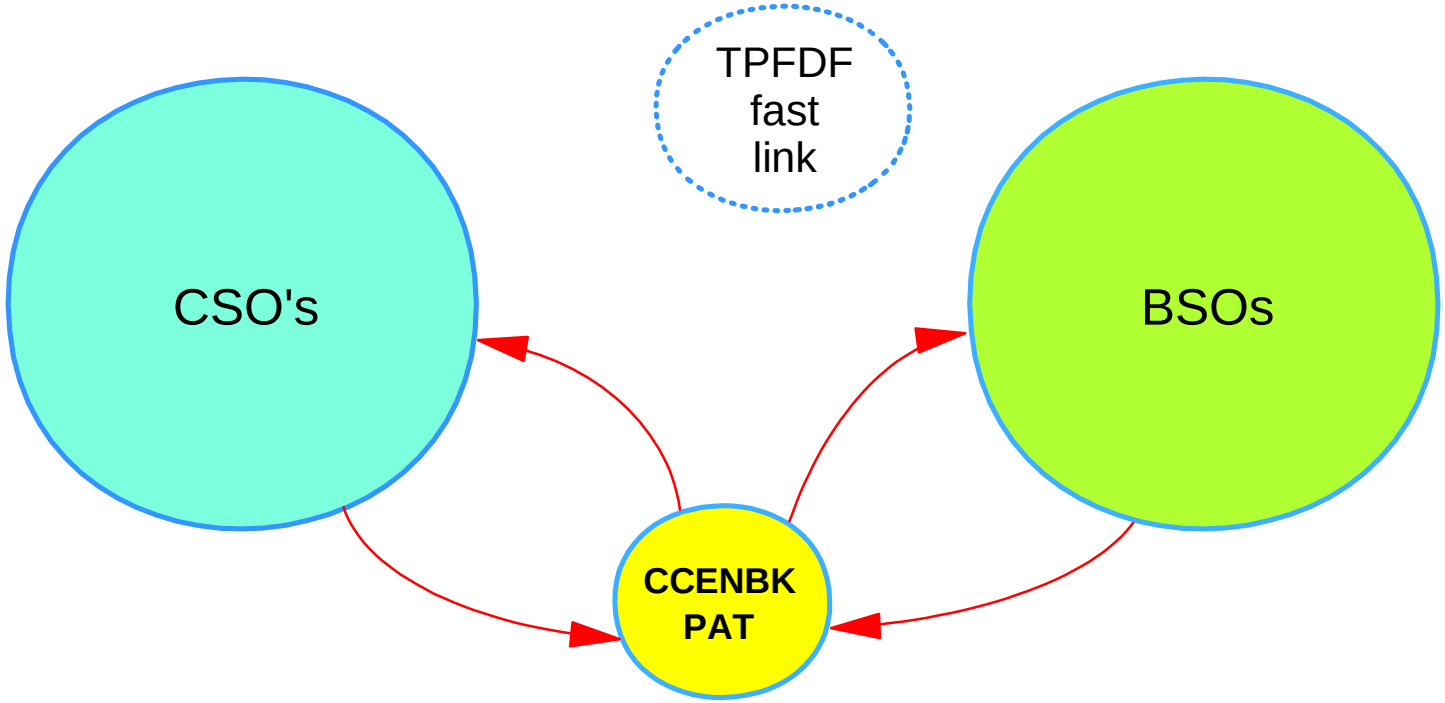
Program Linkage

TPF4.1 linkage review



Program Linkage

z/TPF linkage



Program Linkage Overview

- Linkage stubs are provided for these linkages types:
  - CSO-CSO
  - BSO-BSO
  - CSO-BSO
  - BSO-CSO
- Linkage stubs must also be aware of addressing mode:
  - 64bit-31bit
  - 31bit-64bit
- Additional stubs are provided for:
  - Program trace groups
  - Standard library fast linkage
  - Special library faster linkage

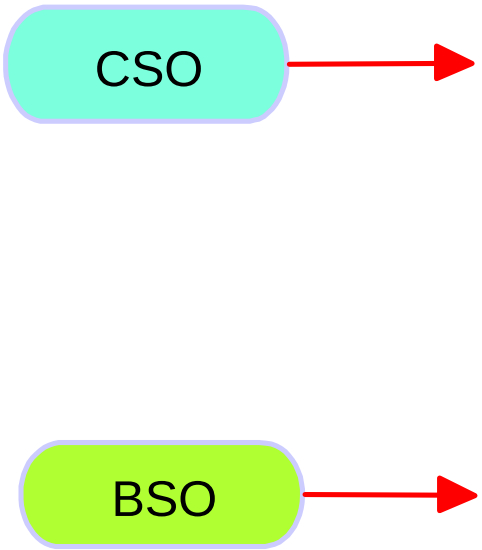
Linkage Types

- Load Time Linkage
- Link time linkage
- Run Time Linkage
- Nonstandard Linkage

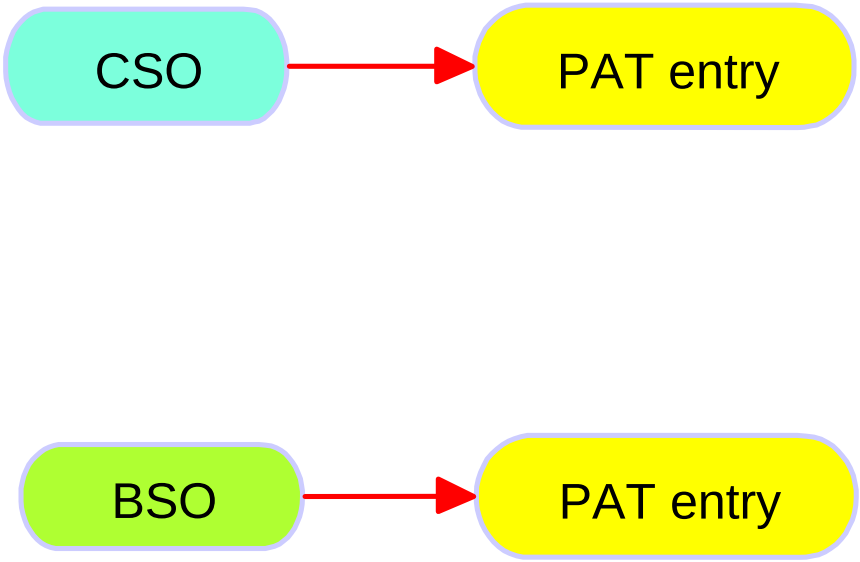
Load Time Linkage

- Normal linkage (external or internal)
- Linkages are resolved at program load time
- Round trip cost for BSO external linkage is 151 instructions

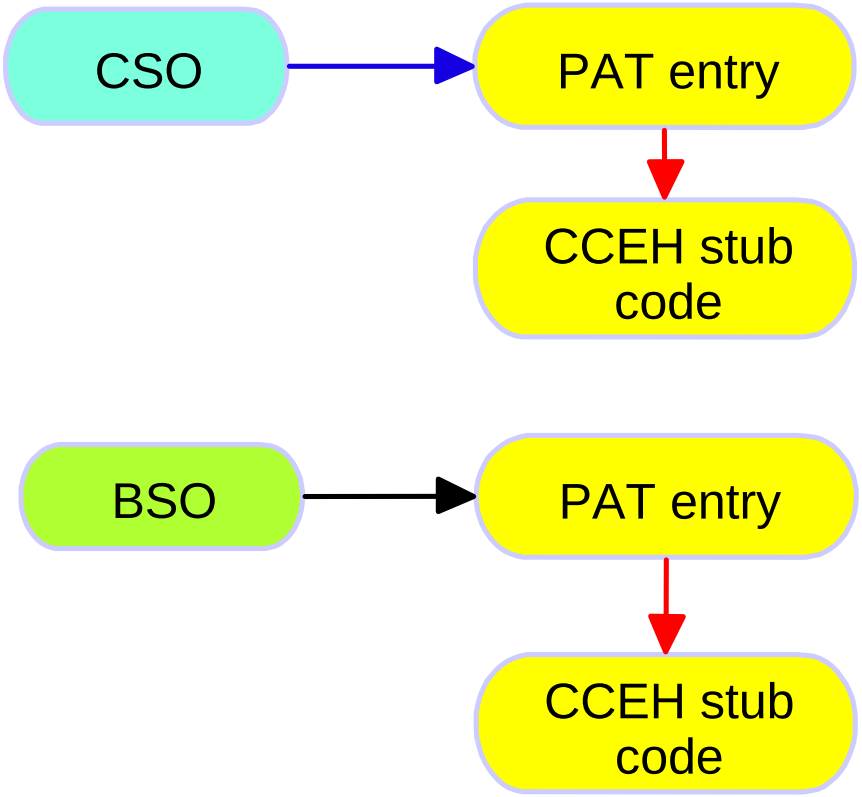
External Linkage



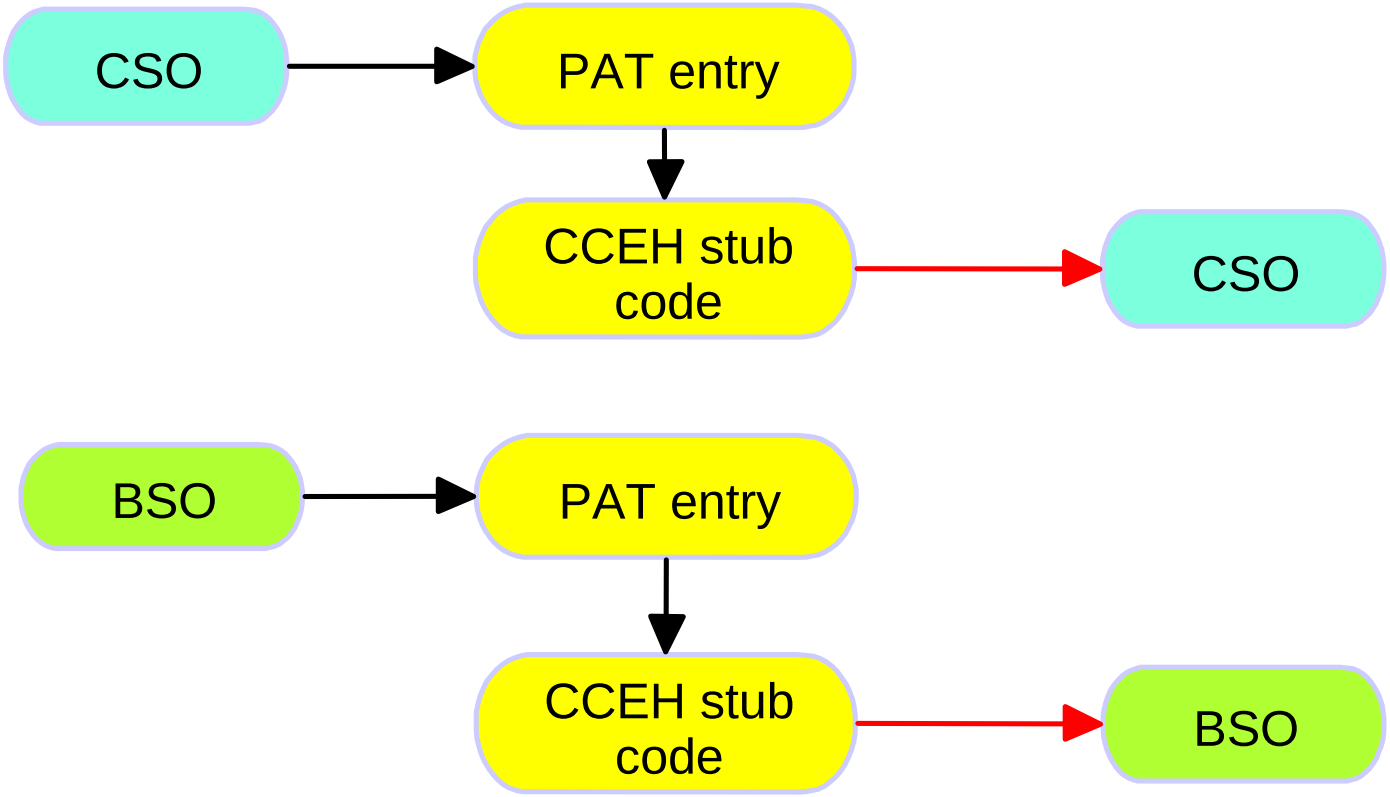
External Linkage



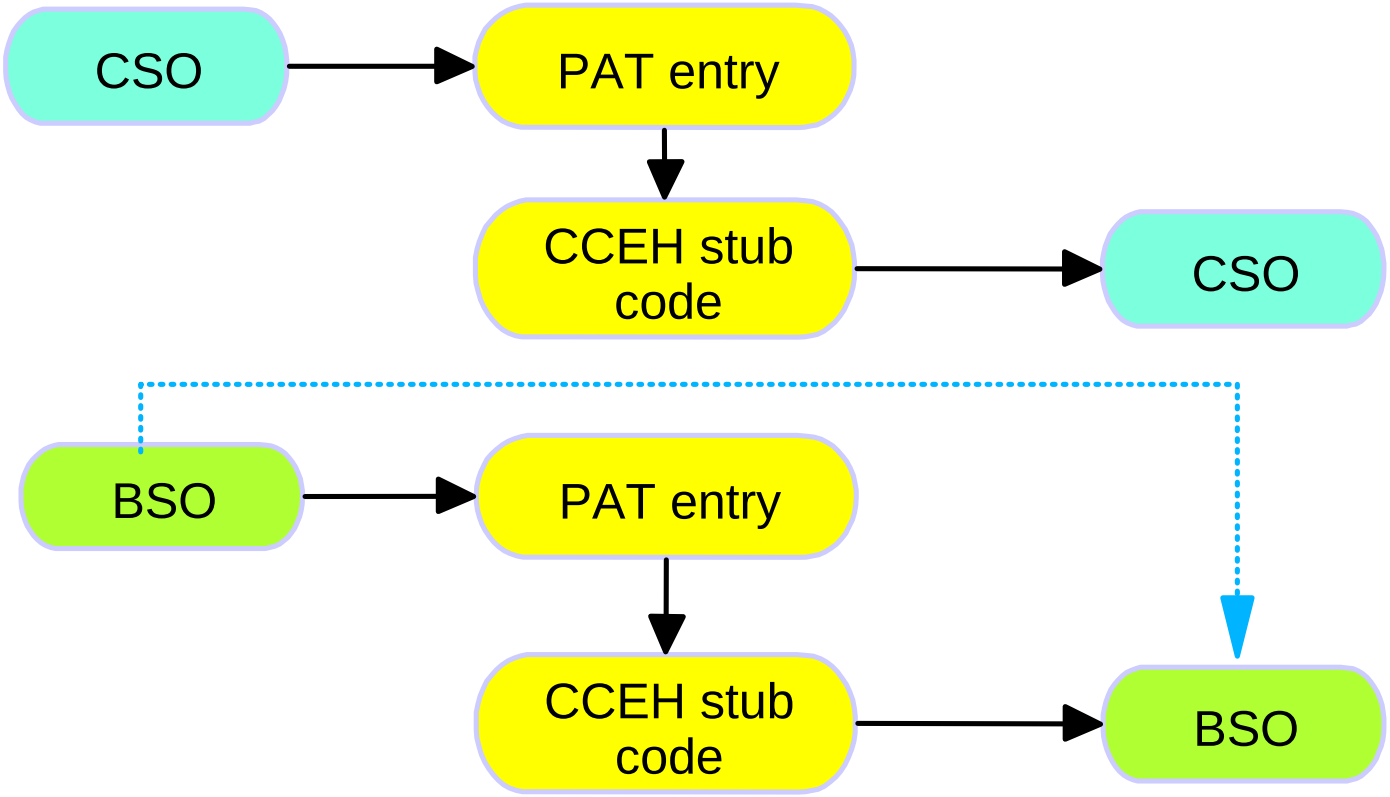
External Linkage



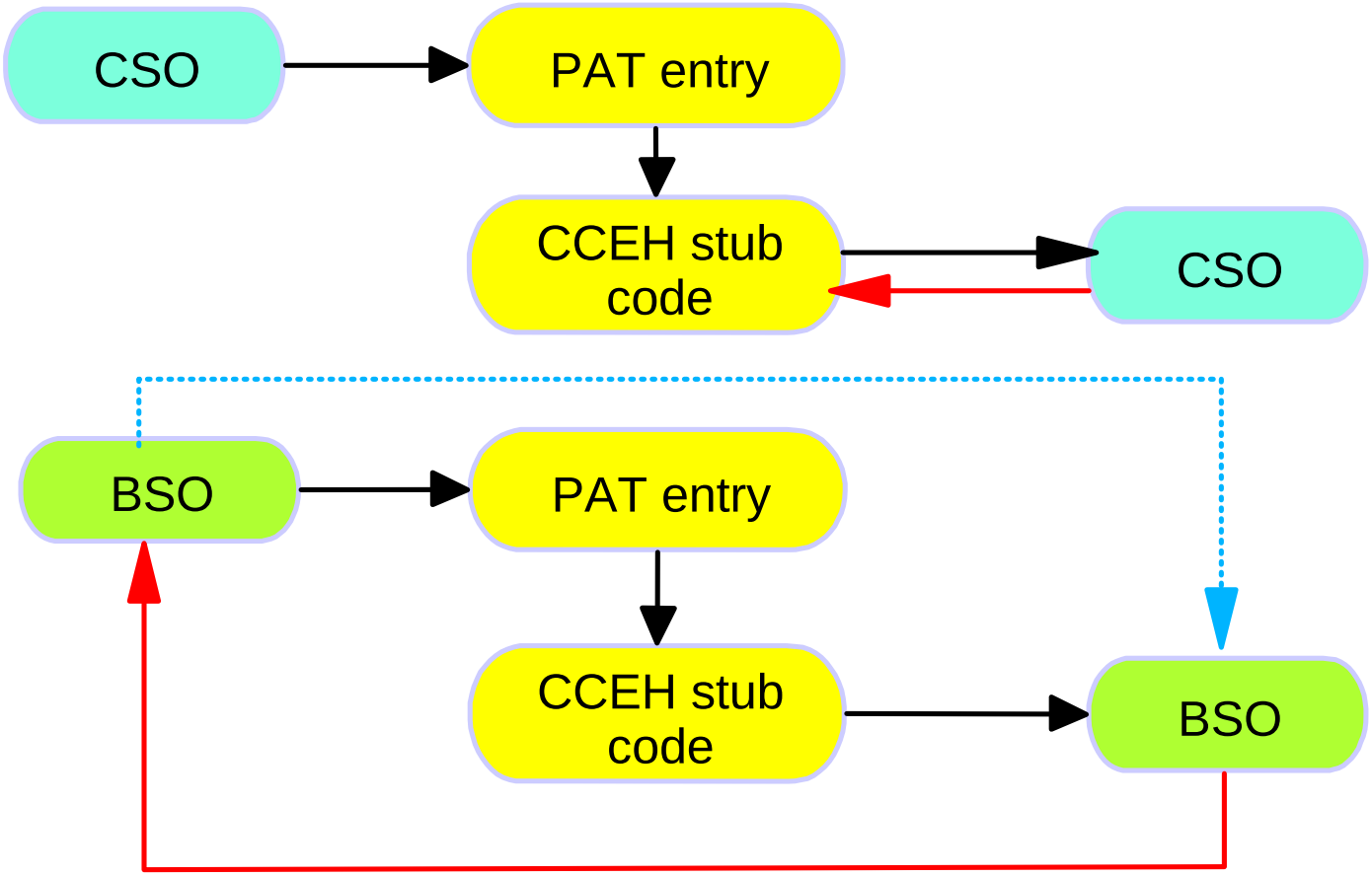
External Linkage



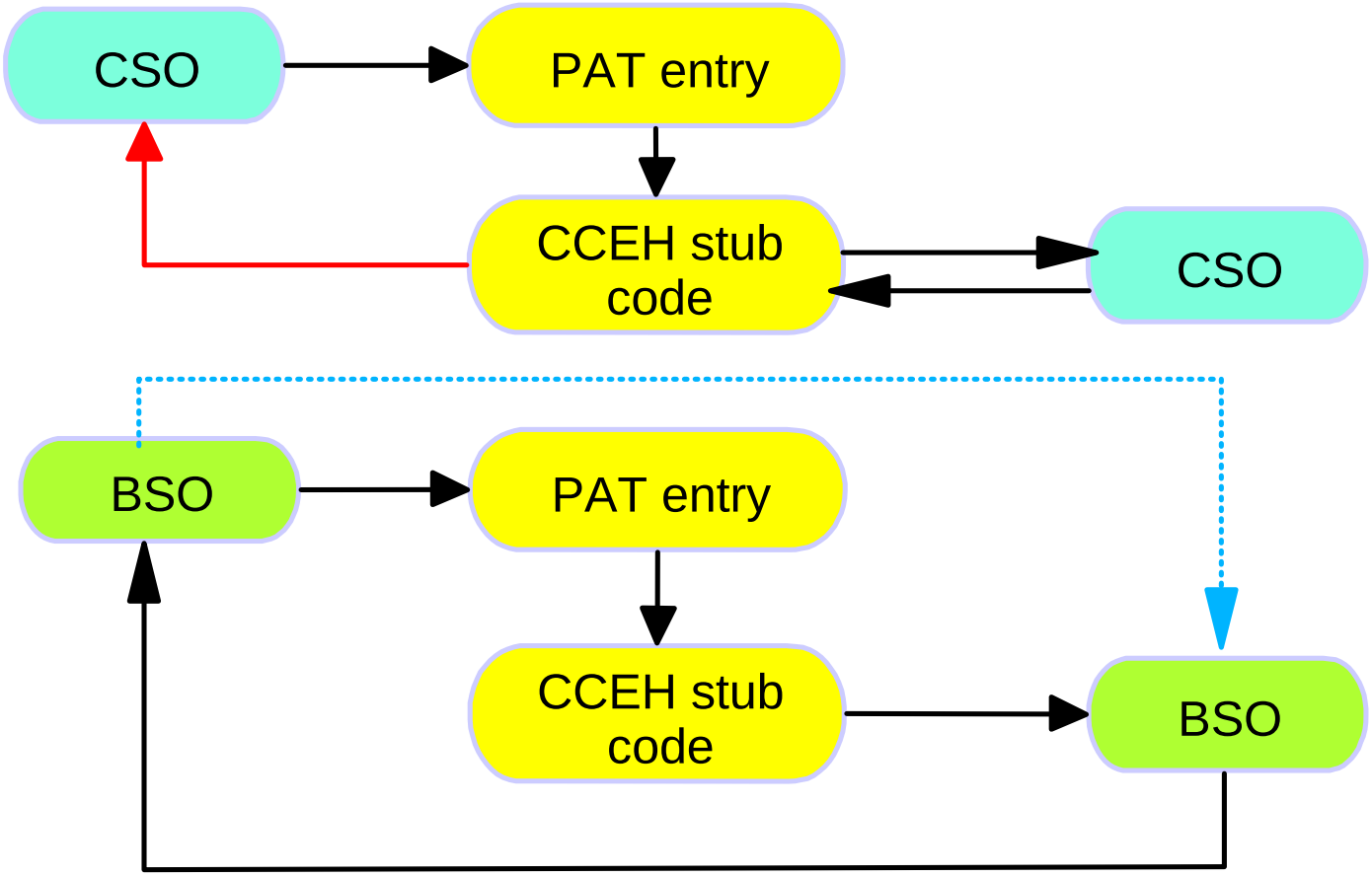
External Linkage



External Linkage



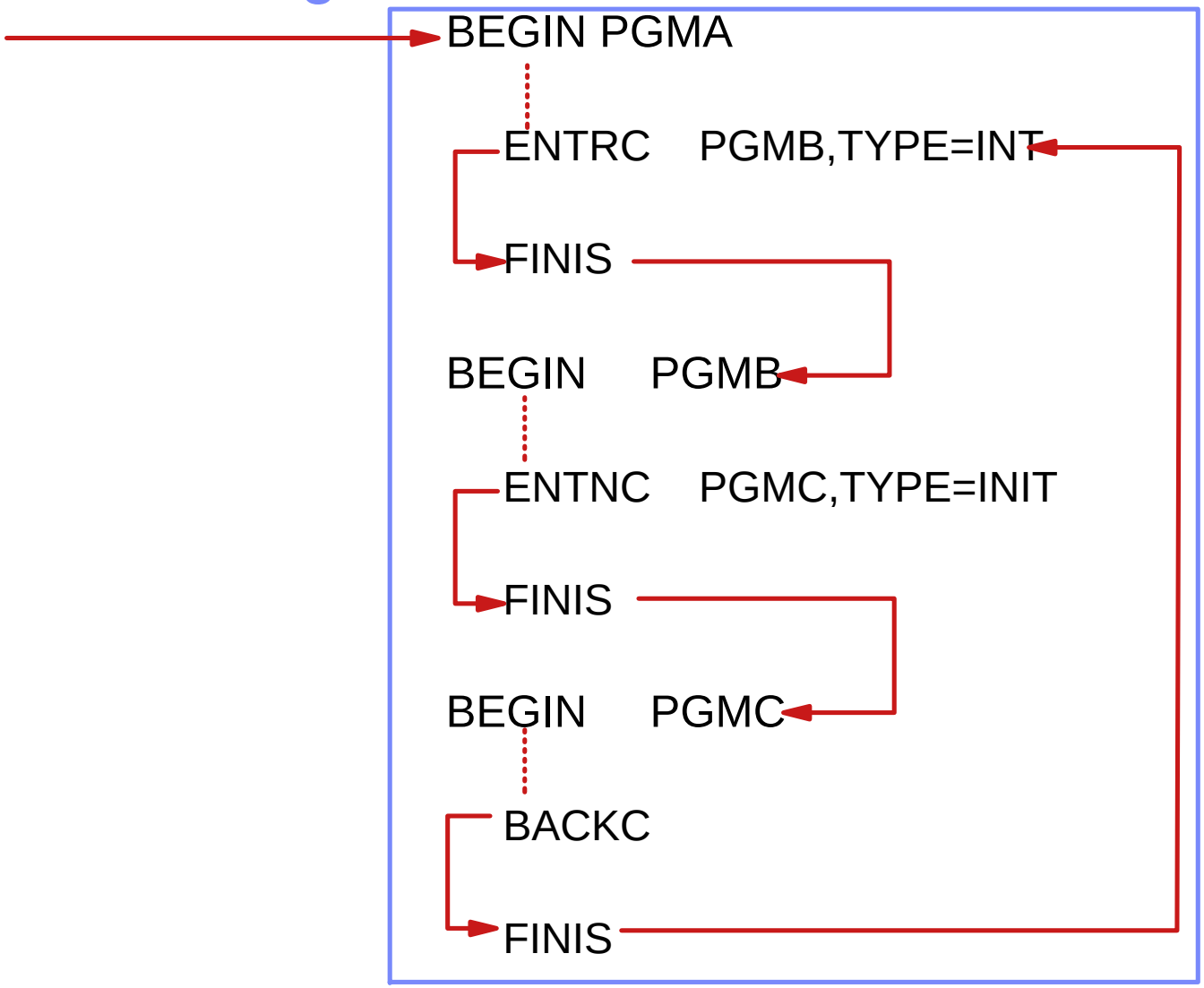
External Linkage



Link Time Linkage

- Special Fast Linkage (internal)
- CSO linkage generated by compiler
- BSO linkage generated by macros
  - BEGIN - generates code to support internal linkage
  - ENTNC TYPE=INT
  - ENTRC TYPE=INT|TPFDF,SAVEREGS=YES|NO (CP usage)
  - BACKC TYPE=TPFDF
  - CLINKC, SLINKC, and RLINKC subroutine linkage
- Round trip cost for BSO internal linkage is 47 instructions

Internal Linkage for BSOs



Run Time Linkage

- Longest linkage (external)
- Linkage resolved at run time
- SO branches to program name hash routine in CCEH
- Hash routine branches to PAT stub
- Used when ENTxC PROGRAM=name is coded

Nonstandard Linkage

- Getting the base address of a program and branching to it.
- Define program as TYPE=DATA
- Use GETPC prog ADDR=R7 to get core address

Trademarks

Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.