

Jądro systemu OS-9 – moduł *kernel*

Podstawowe funkcje jądra systemu:

- zarządzanie zasobami
- zarządzanie pracą procesów
- zarządzanie obsługą zdarzeń specjalnych i przerwań

Realizowane i koordynowane usługi:

- usługi systemowe (np. F\$Link):
 - zarządzanie pamięcią
 - inicjalizacja systemu
 - tworzenie procesów i zarządzanie ich pracą
 - obsługa zdarzeń specjalnych i przerwań
- usługi wejściowo-wyjściowe (np. I\$Open - odczyt, zapis, ...)

Tryby pracy procesora

- **Tryb użytkownika (*user state*)** – normalny tryb wykonywania kodu realizującego proces:
 - ograniczony dostęp do zasobów sprzętowych,
 - zabezpieczenie przed uszkodzeniem danych systemowych (ochrona pamięci),
 - łatwość przenoszenia aplikacji (niezależność od sprzętu).
- **Tryb systemowy (*system state*)** – obsługa usług systemowych, przerwań, zasobów sprzętowych:
 - pełny dostęp do zasobów (maskowanie przerwań, brak ochrony pamięci, uprzywilejowane rozkazy procesora),
 - dostępność wyróżnionych usług systemowych,
 - możliwość rekonfiguracji systemu.

Metody instalowania procedur trybu SYSTEM (dostępne tylko dla użytkowników z grupy "0")

1. Przez umieszczenie modułu **OS9P2** w obrazie ładowanym przy starcie, lub w ROM. Przy restarcie systemu, *kernel* uruchamia ten moduł od **M\$Exec**, co pozwala zainstalować nowe procedury poprzez usługę **F\$SSvc**.
2. Przez system wejścia–wyjścia (za pomocą własnego, pozornego driver-a, z którego można wykorzystać np. **I\$GetStt**).
3. Przez moduł obsługi przerwań programowych TRAP #1..14 (*Trap Handler*).
4. Przez ustawienie bitu *supervisor* w **M\$Attr** modułu programowego, który jest własnością grupy "0".

Wybrane usługi trybu USER

moduły	pamięć	procesy	inne
F\$CRC	F\$AllBit	F\$Alarm	F\$Icpt
F\$DatMod	F\$CpyMem	F\$Chain	F\$RTE
F\$GModDir	F\$DelBit	F\$Event	F\$SetSys
F\$link	F\$Mem	F\$Fork	F\$TLink
F\$Load	F\$SchBit	F\$GPrDBT	F\$CmpNam
F\$SetCRC	F\$SRqCMem	F\$GPrDsc	F\$PrsNam
F\$UnLink	F\$SRqMem	F\$ID	F\$PErr
F\$UnLoad	F\$SRtMem	F\$Send	F\$Gregor
	F\$Trans	F\$Sigmask	F\$Julian
		F\$Sleep	F\$STime
		F\$SPrior	F\$Time
		F\$SSpol	
		F\$SUser	
		F\$UAcct	
		F\$Wait	

Wybrane usługi trybu SYSTEM

moduły	pamięć	procesy	inne
F\$ VModul	F\$Move	F\$Alarm F\$AIIPD F\$AIIPrc F\$AProc F\$DelPrc F\$FindPD F\$IOQn F\$NProc F\$Panic F\$RetPD	F\$IRQ F\$SSvc

Dostęp do usług systemu OS-9

W asemblerze: przez makro: OS9 <kod> realizowane jako:

```
TRAP      #0          ;przerwanie programowe
DC.W      F$Link     ;kod usługi
```

d0.w = kod typu/języka (0 = dowolny)

(a0) = wskaźnik nazwy modułu

Wyniki poprawne:

d0.w = kod typu/języka

d1.w = atrybuty/wersja modułu

(a1) = adres punktu startowego modułu

(a2) = adres początku nagłówka modułu

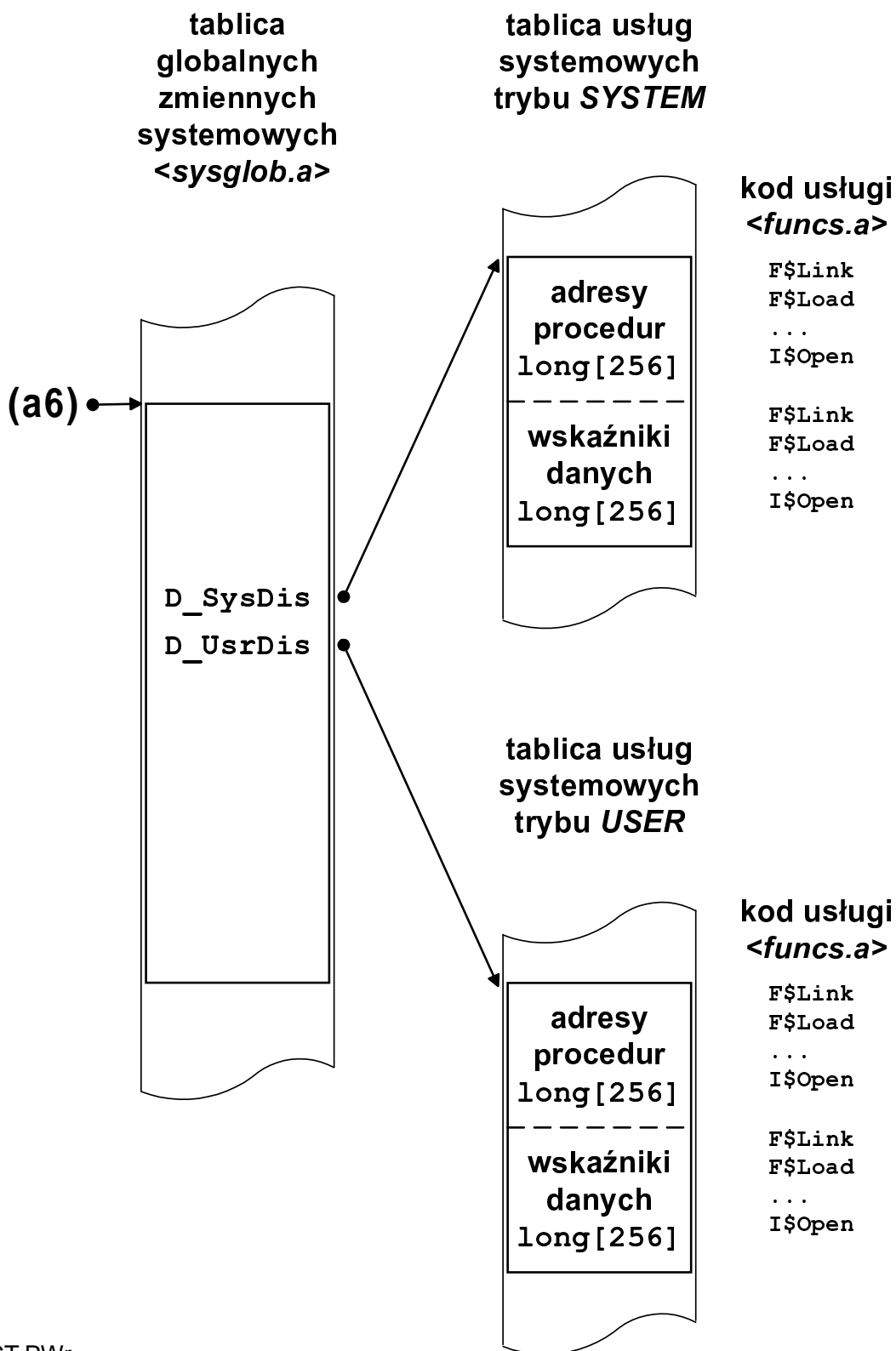
Wyniki przy błędzie:

cc = ustawiony bit Carry

d1.w = kod błędu

W języku C: przez funkcje biblioteczne:

```
#include <module.h>
mh_com *modlink(char *modname, int  typelang);
```



Instalowanie usług systemowych

OS9 F\$SSvc

Parametry:

(a1) = adres tablicy inicjalizacji usługi

(a3) = globalny wskaźnik danych dla usługi

Wynik błędny:

cc = ustawiony bit Carry

d1.w = kod błędu (errno)

Struktura tablicy:

SvcTbl	dc.w	F\$Service	;kod usługi (U+S)
	dc.w	Routine-*-2	;adres procedury
	dc.w	F\$Service+SysTrap	;kod usługi (tylko
	dc.w	SysRoutn-*-2	;adres procedury
	...		
	dc.w	-1	;koniec tablicy

Zasady konstrukcji procedury usługowej

Interpretacja rejestrów na wejściu:

(a3) = wskaźnik danych dla usługi (ustawiony przez $\mathbb{F}\$SSvc$)

(a4) = adres deskryptora bieżącego procesu

(a5) = adres obrazu rejestrów użytkownika

(a6) = adres systemowych danych globalnych

Wyjście poprawne:

cc = ustawiony bit Carry

Wyjście przy błędzie:

cc = zerowy bit Carry

d1.w = kod błędu (errno)