

Získávání nápovědy

Manuálové stránky (Unix/Linux)

Nápověda k prohlížeči	man man
Ovládání: dolů o řádek – o stránku nahoru o řádek – o stránku na začátek, konec hledání ukončení	↓, Enter mezera , PgDown ↑ PgUp Home , End / text q , Q
Výběr specifické úrovně	man n stránka man 3 printf

Ovládání závisí na hodnotě proměnné prostředí \$MANPAGER, implicitně je to program less.

Info stránky

Nápověda k prohlížeči	man info info info
Ovládání: jemný pohyb pohyb po stránkách, kapitolách na začátek, konec návrat do nevyššího menu do vyšší úrovně menu předchozí položka menu následující položka menu hledání ukončení	Enter , ←, →, ↑, ↓ Mezera , Backspace , Del , PgUp , PgDown Home , End t u p n / text q , Q

Užitečné programy

whatis slovo	Prohledá databázi manuálových stránek a vypíše krátkou informaci o zadané stránce. Vyhledává stránky stejného jména jako zadané slovo.
apropos slovo	Jako whatis, ale vyhledá všechny stránky, které ve svém jméně obsahují zadané slovo.

Standardní knihovny

<ctype.h> <errno.h> <float.h> <limits.h>
<math.h> <stdbool.h> <stdio.h> <stdlib.h>
<string.h> <time.h>

Testování znaků <ctype.h>

znak nebo číslice?	isalnum(c)
znak?	isalpha(c)
řídící znak?	isctr1(c)
čísllice?	isdigit(c)
malé písmeno?	islower(c)
tisknutelný znak? (včetně mezery)	isprint(c)
bílý znak?	isspace(c)
velké písmeno?	isupper(c)
hexa znak? (0..F)	isxdigit(c)
na malé písmeno	tolower(c)
na velké písmeno	toupper(c)

Operace se stringy <string.h>

s, cs, ct jsou stringy

délka stringu	strlen(s)
kopíruje cs do s – max. prvních n znaků	strcpy(s, cs)
připojí cs za s – prvních n znaků	strncpy(s, cs, n)
porovná cs a ct – max. prvních n znaků	strcat(s, cs)
najde první výskyt znaku c v cs – poslední výskyt znaku c v cs	strncat(s, cs, n)
	strcmp(cs, ct)
	strncmp(cs, ct, n)
	strchr(cs, c)
	strrchr(cs, c)

b, cb, ct jsou bloky paměti

kopie n bytů z cb do b – b a cb se mohou překrývat	memcpy(b, cb, n)
porovná n bytů z cb s ct	memcmp(b, cb, n)
najde první znak c v prvních n bytech cb	memcmp(cb, ct, n)
vyplní prvních n bytů b hodnotou c	memchr(cb, c, n)
	memset(b, c, n)

Vstup/Výstup <stdio.h>

Standardní vstup/výstup

standardní vstup	stdin
standardní výstup	stdout
standardní chybový výstup	stderr
konec souboru (konstanta)	EOF
načti znak (int) ze stdin	getchar()
vytiskni znak na stdout	putchar(chr)
vytiskni formátovaná data do stdout – do stringu, pole znaků s	printf("format", arg1, ...) sprintf(s, "format", arg1, ...)
čti formátovaná data ze stdin – ze stringu s	scanf("format", &prom1, ...) sscanf(s, "format", &prom1, ...)

Souborový vstup/výstup

ukazatel na proměnnou značící soubor	FILE *f;
otevření souboru	f = fopen("jméno", "mód"); módy: r (read), w (write), a (append)
zavři soubor	fclose(f)
při chybě vračí != 0	ferror(f)
načti znak	fgetc(f)
zapiš znak c	fputc(c, f)
formátovaný zápis	fprintf(f, "format", arg1, ...)
čtení ze souboru	fscanf(f, "format", &arg1, ...)

čti řádek do s (< max znaků)	fgets(s, max, f)
zapiš string s	fputs(s, f)

Formátovaný výstup (printf)

formát	argument	co se vypíše
%d %i	int	znaménkové desítkové číslo
%o	unsigned int	osmičkové číslo
%u	unsigned int	desítkové číslo
%x %X	unsigned int	hexa číslo
%ld	long	dlouhé desítkové číslo
%lo %lu	unsigned long	dlouhé číslo
%lx		

%15d	int	zarovná desítkové číslo na 15 míst mezerami, pokud má číslo více číslic, zarovnání se ignoruje
%e %E	double	exponenciální formát reálného čísla
%f %F	double	decimální formát reálného čísla
%g %G	double	jako %e nebo %f podle toho, co lépe vyplní prostor
%10.5f	double	zarovnání na 10 míst, číslo se zaokrouhlí na 5 desetinných míst
%15.2g	double	15 míst celkem, 2 desetinná místa, formátem %f nebo %e podle toho, co se sem lépe vejde
%.4f	double	4 desetinná místa, nezarovnáno
%s	char *	textový řetězec
%c	char	znak
%%	-	vypíše znak %
\n \t	-	nový řádek, tabulátor

Příklad:

```
double a = 657.73578, b = 55.4152565;
printf("%8.2f + %8.2f = %8.2e\n", a, b, a+b);
```

Výsledek:

```
657.74 + 55.42 = 7.13e+02
```

Formátovaný vstup (scanf)

formát	argument	co se načte
%d %ld	int*, long*	desítkové číslo
%o %lo	int*, long*	osmičkové číslo
%x %lx	int*, long*	šestnáctkové číslo
%h	short*	short
%f %lf	float*, double*	reálné číslo
%c	char*	jediný znak
%s	char*, char[]	textový řetězec (slovo)
.*d	-	přeskočí desítkové číslo

Příklad:

```
int i; float x; char name[50];
scanf("%d %f %.*s", &i, &x, name);
```

vstup: 56789 0123 45a72

výsledek: i = 56; x = 789.0; name = "45";

Standardní funkce <stdlib.h>

absolutní hodnota int n	abs(n)
absolutní hodnota long n	labs(n)
pseudonáhodné číslo <0, RAND_MAX)	rand()
nastav náhodné semínko	srand(n)
konec programu	exit(kód);
spustí systémový příkaz	system("dir");

Konverze

string s na double	atof(s)
string s na int	atoi(s)
string s na long	atol(s)
string s na double	strtod(s, &endp)
string s na long	strtoul(s, &endp, b)
string s na unsigned long	strtoul(s, &endp, b)

Alokace paměti

alokace	if ((ptr = malloc(size)) == NULL) {ošetření chyby}
uvolnění paměti	free(ptr); //free(NULL) není chyba!

Funkce malloc vrací hodnotu NULL v případě, že při alokaci došlo k nějaké chybě. Nejčastější chybou je nedostatek volné paměti. Slušná aplikace provádí testování chyby při KAŽDÉM volání funkce malloc!

Matematické funkce <math.h>

Při použití matematických funkcí je nutné (v prostředí POSIX/UNIX) aplikaci sestavit s matematickou knihovnou. Při překladu je nutné spouštět překladač s parametrem -lm (malé l jako „link“, malé m jako „math library“).

Všechny funkce z matematické knihovny pracují s reálnými datovými typy (double).

trigonom. fce.	sin(x), cos(x), tan(x)
inverzní trig. fce.	asin(x), acos(x), atan(x)
arctan(y/x)	atan2(y, x)
expon. a logaritmické fce.	exp(x), log(x), log10(x)
mocnina	pow(x, y)
odmocnina	sqrt(x)
zaokrouhlení	round(x), ceil(x), floor(x)
absolutní hodnota	fabs(x)

Doporučené nastavení překladače GCC

Pro začátečníky je důležité, aby překladač odchytil co největší množství chyb, a aby bylo možné přeložený program krokodýlovit pomocí debuggeru.

příklad	gcc -std=c99 -Wall -pedantic -W -g -o projekt projekt.c
debugger	ddd projekt

Přehled knihoven jazyka C. Autor: David Martinek ©2004–2007. Tento dokument neobsahuje úplný popis standardních knihoven jazyka C. Je to pouze přehled základních funkcí, které jsou potřeba pro tvorbu projektů pro kurz Základy programování na FIT VUT v Brně. Pripomínky pošlete na martinek@fit.vutbr.cz nebo na adresu: David Martinek, Fakulta Informačních Technologií, VUT v Brně, Božetěchova 2, 612 22, Brno.

Tento dokument je šířen pod licenci GNU FDL (plné znění na <http://www.gnu.org>).